Naval Research Laboratory
Stennis Space Center, MS 39529-5004

NRL/MR/7320--01-8271

Software Design Description for the Advanced Circulation Model for Shelves, Coastal Seas, and Estuaries (ADCIRC)

CHERYL ANN BLAIN

Ocean Dynamics and Prediction Branch
Oceanography Division

KIMBERLY A. KELLY

Planning Systems Incorporated
Slidell, LA

December 21, 2001

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

1. REPORT DATE *(DD-MM-YYYY)*	2. REPORT TYPE	3. DATES COVERED *(From - To)*
December 21, 2001		

4. TITLE AND SUBTITLE

Software Design Description for the Advanced Circulation Model for Shelves, Coastal Seas, and Estuaries (ADCIRC)

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)

Cheryl Ann Blain and Kimberly A. Kelly*

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Naval Research Laboratory
Oceanography Division
Stennis Space Center, MS 39529-5004

8. PERFORMING ORGANIZATION REPORT NUMBER

NRL/MR/7320--01-8271

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)

Office of Naval Research
800 North Quincy St.
Arlington, VA 22217-5660

10. SPONSOR / MONITOR'S ACRONYM(S)

11. SPONSOR / MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION / AVAILABILITY STATEMENT

Approved for public release; distribution is unlimited.

13. SUPPLEMENTARY NOTES

*Planning Systems Incorporated, 115 Christian Lane, Slidell, LA

14. ABSTRACT

The purpose of this Software Design Description (SDD) is to describe the software design and code of the Advanced Circulation Model for Shelves, Coastal Seas, and Estuaries (ADCIRC). The structure of the code is outlined using flow charts with details of the input and output files included. Detailed descriptions of the programs, subprograms, and common blocks are provided. Information regarding specification of input files and the implementation of the setup code are also included. Specifics on the model theory and a detailed user manual with advice on parameter selection are referenced.

15. SUBJECT TERMS

Finite element model, tide/surge model, tidal height prediction

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			Cheryl Ann Blain
Unclassified	Unclassified	Unclassified	UL	125	19b. TELEPHONE NUMBER *(include area code)* 228-688-5450

Standard Form 298 (Rev. 8-98)
Prescribed by ANSI Std. Z39.18

TABLE OF CONTENTS

TABLE OF FIGURES

1.0 SCOPE

1.1 Identification

The Computer Software Configuration Item (CSCI), identified as the Advanced Circulation Model for Shelves, Coastal Seas, and Estuaries (ADCIRC) is a hydrodynamic model which implements the generalized wave-continuity equation (GWCE) and momentum balance equations to solve for tidal elevation and current velocity within coastal waters. ADCIRC uses the finite element method, which allows for high resolution of shoreline geometry and increased refinement near shallow, coastal areas and near regions of swiftly changing bathymetry.

ADCIRC can be run in either a two-dimensional depth-integrated (2DDI) form or in 2 alternative three-dimensional forms. This document will describe the two-dimensional depth-integrated form (2DDI). A Cartesian or spherical coordinate system is used for the horizontal solution of the generalized shallow water equations. Elevation is calculated from the solution of the generalized wave-continuity equation (GWCE). This equation can be solved using either a consistent or a lumped mass matrix and a partially implicit or a fully explicit time stepping procedure. Velocity is obtained from the solution of the momentum equations whose specific form depends on whether the model is run in 2DDI or 3D.

ADCIRC-2DDI also computes tidal harmonic constituents during the course of the run using least square analysis routines. This advantage allows the user to avoid writing out long time series for post-processing.

1.2 Document Overview

The purpose of this Software Design Description (SDD) is to describe the software design and code of the Advanced Circulation Model for Shelves, Coastal Seas, and Estuaries (ADCIRC-2DDI). It includes flow charts and descriptions of the programs, subprograms, and common blocks. This document will form the standard documentation used to assist the War Fighting Support Center (WSC).

2.0 REFERENCED DOCUMENTS

2.1 General Technical Documentation

Leuttich, R.A., Jr., J.J. Westerink, and N.W. Scheffner. 1992. An advanced three dimensional circulation model for shelves, coasts, and estuaries, Report 1, Theory and Methodology of ADCIRC-2DDI and ADCIRC-3DL. Department of the Army, Technical Report DRP-92-6.

Westerink, J.J., C.A. Blain, R.A. Leuttich, Jr., N.W. Scheffner. 1994. ADCIRC: An advanced three dimensional circulation model for shelves, coasts, and estuaries, Report 2, User's manual for ADCIRC-2DDI. Department of the Army, *Technical Report DRP-92-6.*

Manuscript approved October 18, 2001.

3.0 CSCI-WIDE DESIGN DECISION

The ADCIRC-2DDI model uses a sequential access file format for all files except hot start output files (fort.67, fort.68) and output files such as fort.61, fort.62, fort.63, fort.64, fort.74, fort.71, and fort.73 that have been specified by the user as direct access (i.e. NOUTGE = 2, NOUTGU = 2, etc.).

4.0 CSCI ARCHITECTURAL DESIGN

4.1 CSCI Components

a) ADCIRC can be divided into three main software units: CSC ADCSETUP, CSC ADCIRC, and CSC SOLVER.

b) CSC ADCSETUP- This program is an interactive pre-processor which completely automates the configuration of ADCIRC for a specific application based on information extracted from the grid file (fort.14) and the parameter input file (fort.15). Modifications to the ADCIRC code address dimensioning, input consistency, forcing, and computer platform.

CSC ADCIRC- This program consists of 39 subroutines and 13 common blocks that in combination produce solutions for water elevation and depth-averaged velocity found in coastal regions due to tides and atmospheric forcing. Tidal harmonic constituents are computed throughout the execution of the program, if desired.

CSC SOLVER- A direct banded matrix solver (programs SGBCO and SGBSL from LINPACK) or one of 7 different iterative solvers (from ITPACKV 2D) are used to solve the GWCE. However, if the GWCE is set up using a lumped, fully explicit formulation, no matrix solver is necessary. The selected solver must be compiled and linked to ADCIRC.

c) The ADCIRC model was originally developed by Drs. R. A. Leuttich and J. J. Westerink in 1992 with the support of the Army Corps of Engineers, and has undergone numerous revisions and additions since this form, Version 33.05.

d) The ADCIRC model takes up about 902 KB of disk space. This number includes the files adcsetup.f, adc33_05.for, and srcv2d.for. It does not include additional space that is required for the input files. The size of the input files varies depending on the size of the problem (i.e. fort.14), wind input option (i.e. fort.22), as well as other external data options specified by the user (i.e. within fort.15).

4.2 Concept of Execution

ADCIRC is a platform independent code that may be run on PC systems, UNIX systems, or all HPC platforms. ADCIRC can be forced with elevation boundary conditions (harmonic tidal

constituents or elevation time series), normal flow boundary conditions, external and internal barrier flow boundary conditions, surface stress boundary conditions (wind), atmospheric pressure, tidal potential, or earth load/self-attraction tide.

Prior to execution of the ADCIRC program, the source code pre-processor adcsetup.f must be run. This reads the UNIT 14 and 15 ADCIRC input files and prompts the user for additional information. Based on this input, the pre-processor sets the dimensions for all variables and optimizes the code for the specific problem and computer to be used.

Execution of this CSCI begins by prompting the user for the following information:

1) Name of the UNIT 14 input file.
2) Name of the UNIT 15 input file.
3) Implementation of a lumped GWCE matrix foundation, yes or no.
4) Type of computer on which ADCIRC will be run (scalar or vector). If scalar, the make of the computer (SUN or other).
5) Type of compiler, 32-bit (e.g. SUN, CONVEX, IBM RS 6000 etc.) or 64-bit (e.g. CRAY, YMP, DEC ALPHA computers).
6) Name of the initial ADCIRC-2DDI FORTRAN source code file.
7) Name of the modified ADCIRC-2DDI FORTRAN source code file that is specific to the problem and computer on which ADCIRC will be run.

Once the pre-processor has run, the output production source code is compiled and linked to the solver. As configured and implemented at the WSC, an iterative solver is specified by the UNIT 15 input file, requiring that ITPACKV 2D package of subroutines ("srcv2d.for") be linked to the source code during compilation.

For a cold start model setup, time-dependent boundary forcing conditions (specified in UNIT 15) and initial wind stress and pressure forcing are read in to the program. The wind and pressure data are specified by the user. ADCIRC supports direct reads from four sources of wind products: 1) outdated FNMOC wind velocity data, 2) PBL/JAG wind model output velocities, 3) NWS AVN model MET files, and 4) NWS ETA-29 MET files. None of these sources are utilized within the WSC. Instead, Navy meteorological products are processed to interface with the ADCIRC fort.22 format under the NWS = 2 option. In addition, initial wet/dry interface nodes are determined and header information is written to output files. Lastly, matrices for harmonic analysis are intialized.

For a hot start model setup, the user must locate the most recently completed hot start file (UNIT 67 or UNIT 68) and set the variable IHOT in UNIT 15 equal to 67 or 68. The program will then locate the appropriate positions in the forcing/output files and re-initialize the model variables.

Prior to time stepping, flags and coefficients used within the time stepping loop are set. Then, in the time stepping procedure, the elevation and velocity are calculated at each node for each time step according to the GWCE and Momentum equations. The results are then written to either global (all nodes) or station (pre-selected nodes) output files. The harmonic analysis matrix and load vectors are also updated. Harmonic analysis completes the

execution of ADCIRC-2DDI by solving the harmonic analysis matrices using a least squares approach and writing the solution to output files. A flow diagram illustrating the basic logic underlying the operation of ADCIRC-2DDI is shown in Figure 4.2-1.

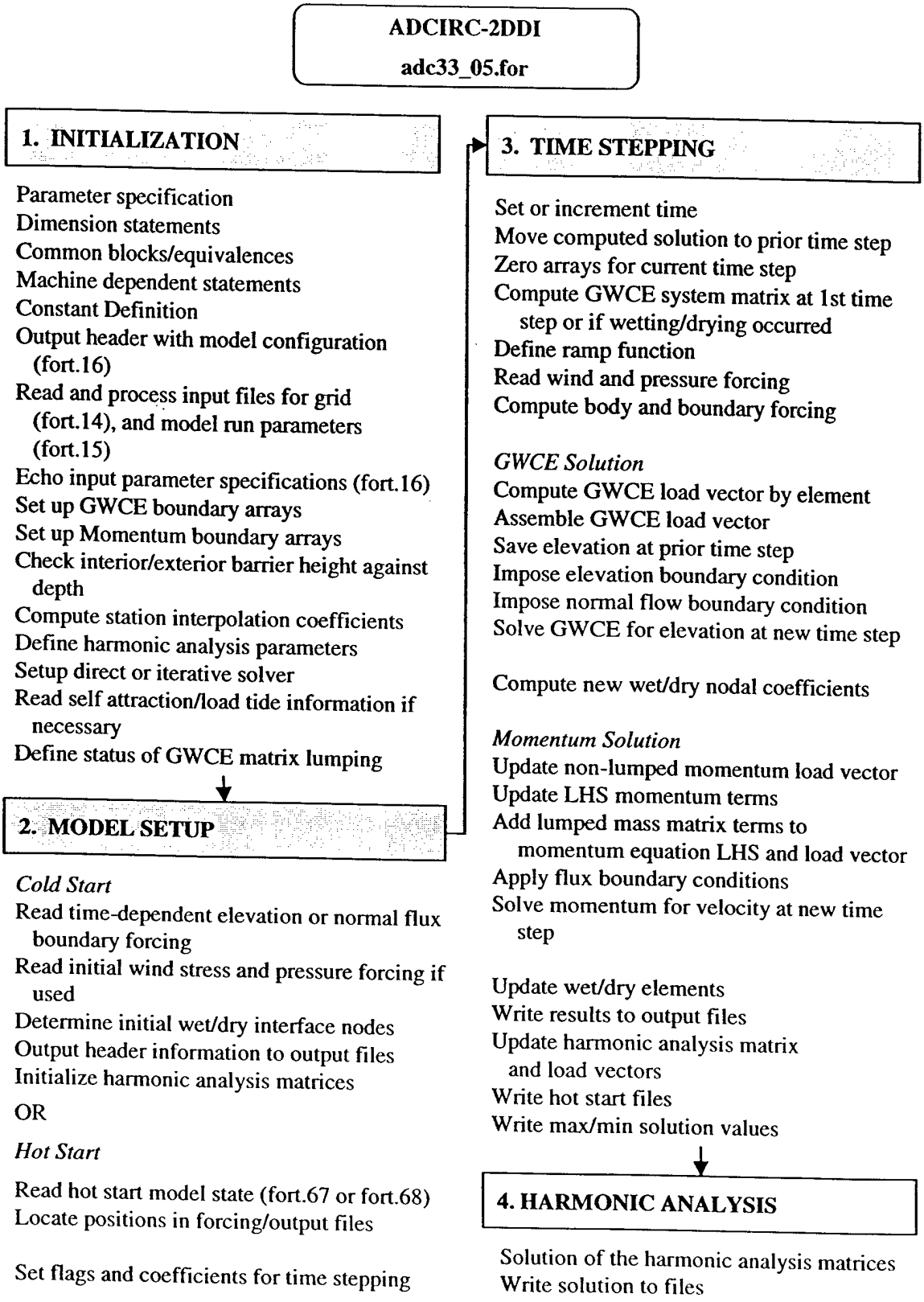

Figure 4.2-1. Flowchart illustrating the execution of the ADCIRC-2DDI model.

4.3 Interface Design

4.3.1 Interface Identification and Diagrams

The only Navy standard ADCIRC external interfaces are the input and output files.

There are two standard input files that must be supplied in order to run ADCIRC-2DDI. These are UNIT 14 and UNIT 15, defined by ADCIRC as "fort.14" and "fort.15". UNITS 19, 20, 21, 22, 200, and 67/68 are supplemental files that may or may not be used under specified conditions.

Table 4.3-1. Description of input files required for running ADCIRC-2DDI.

File	Description
UNIT 14	Finite element grid and boundary information file.
UNIT 15	Input file that defines the majority of input parameters necessary for running the code.
UNIT 19	Non-periodic time-varying elevation boundary conditions at "elevation specified" boundary nodes. This file is only used when an "elevation specified" boundary condition has been specified and NBFR = 0 (see description of UNIT 15 file for NBFR.).
UNIT 20	Non-periodic time-varying normal flow/UNIT width values at "specified normal flow" boundary nodes. This file is only used when a "specified normal flow" boundary condition has been specified and NFFR = 0 (see description of UNIT 15 file for NFFR.).
UNIT 21	Spatially variable friction values given at the nodes. This file is only read in when NWP has been specified equal to 1 in the UNIT 15 input file.
UNIT 22	Wind stress and barometric pressure information file. Read in if NWS = 1 thru 4 in the UNIT 15 input file.
UNIT 23	Self-attraction/load tide information is read in for each constituent at each node in the grid in UNIT 23. When NTIP = 2, both tidal potential and self attraction/load tide functions are used.
UNITS 200, 206, 212, 218, 224, 230, 236, 242, 248.....	Wind stress and barometric pressure information files at 6 hour intervals. Read in if NWS = 10 in the UNIT 15 input file.
UNITS 200, 201, 202, 203, 204, 205, 206, 207, 208.....	Daily wind stress and barometric pressure information containing data at 3 hr. intervals. Read in if NWS = 11 in the UNIT 15 input file.
UNIT 67 or 68	Hot start input files.

There are one standard and 16 supplemental output files. The supplemental files are activated by input parameters specified in the UNIT 15 input file.

Table 4.3-2. Description of output files.

File	Description
UNIT 16	Output file which echo prints model configuration, input parameters, grid data, and provides some runtime information. Error messages regarding parameter specification and/or usage are also contained in this file.
UNIT 6	Screen output. Provides limited run time information as well as warnings.
UNIT 33	Diagnostic output from ITPACKV 2D if an iterative solver is specified.
UNIT 51	Harmonic constituent elevation amplitudes and phases at specified elevation recording station coordinates (ASCII).
UNIT 52	Harmonic constituent velocity component amplitudes and phases at specified velocity recording station coordinates (ASCII).
UNIT 53	Harmonic constituent elevation amplitudes and phases at all nodes (ASCII).
UNIT 54	Harmonic constituent velocity component amplitudes and phases at all nodes (ASCII).
UNIT 55	Comparison between the mean and variance of the time series generated by the model and the mean and variance of a time series resynthesized from the computed harmonic constituents. This gives an indication of how complete the harmonic analysis was.
UNIT 61	Time series elevation values at specified elevation recording station coordinates (ASCII or binary).
UNIT 62	Time series velocity component values at specified velocity recording station coordinates (ASCII or binary).
UNIT 63	Time series elevations at all nodes (ASCII or binary).
UNIT 64	Time series velocity component at all nodes (ASCII or binary).
UNIT 67	Hot start output file 1 = fort.67 (binary).
UNIT 68	Hot start output file 2 = fort.68 (binary).
UNIT 71	Time series concentration values at specified concentration recording stations.
UNIT 73	Time series concentrations at all nodes (ASCII or binary).
UNIT 74	Time series of wind stress at all nodes (ASCII or binary).

The interfacing and operational environment for ADCIRC-2DDI, which demonstrates the relationship between the components of the system and associated files, is illustrated in Figure 4.3-1.

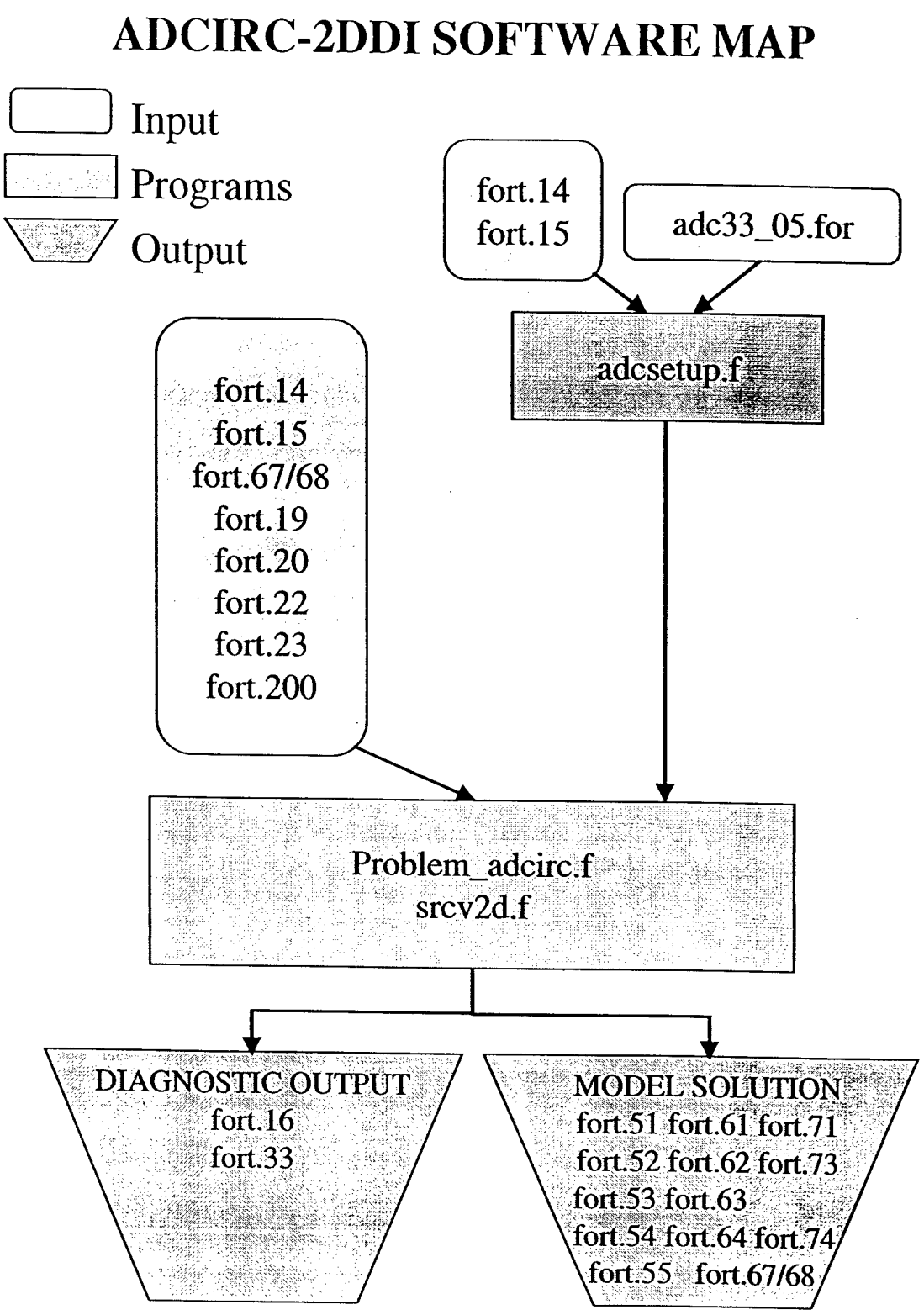

Figure 4.3-1. Flow diagram illustrating the relationship between software components of the ADCIRC-2DDI model. Problem_adcirc.f is the ADCIRC code configured for a specific problem.

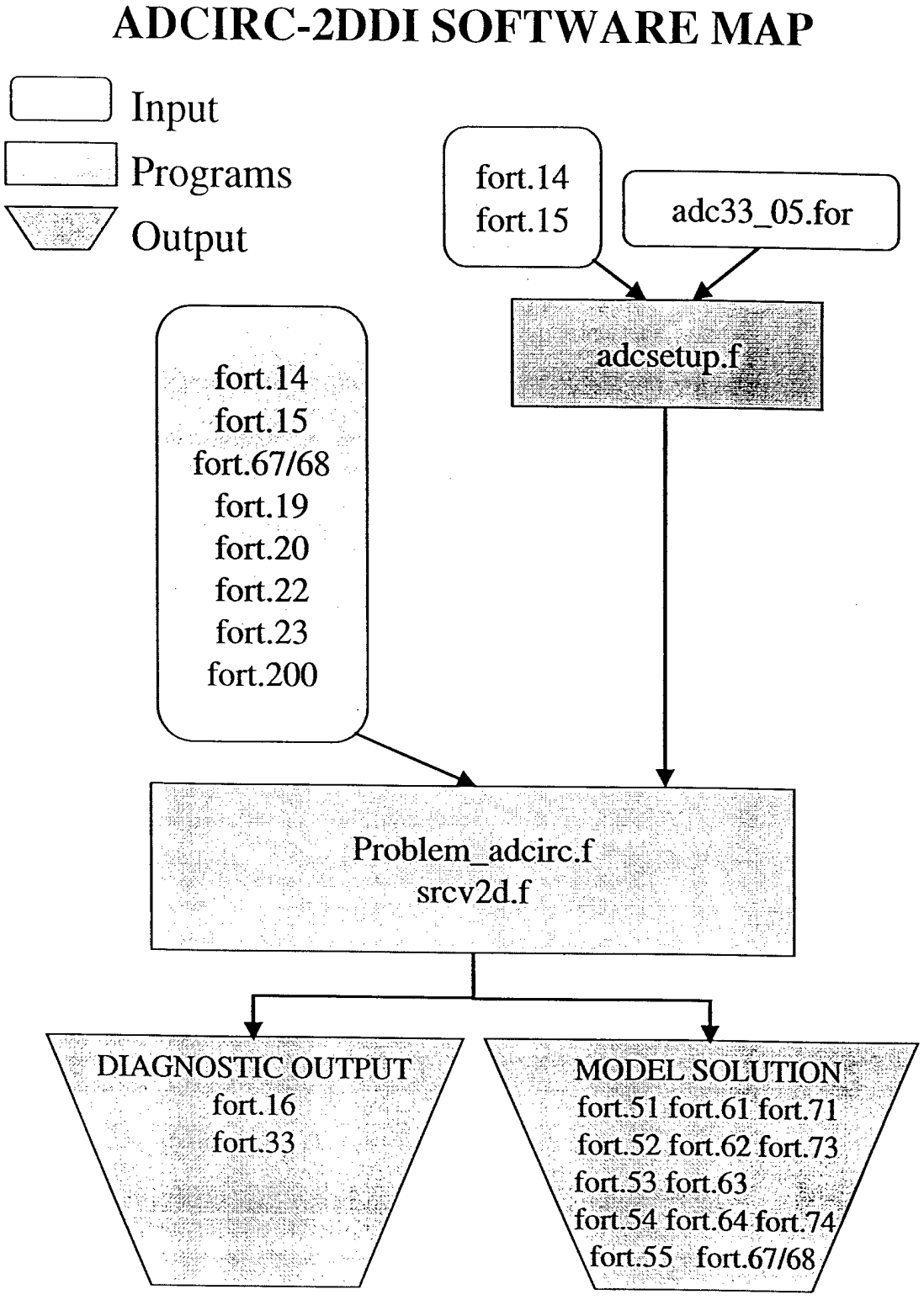

Figure 4.3-1. Flow diagram illustrating the relationship between software components of the ADCIRC-2DDI model. Problem_adcirc.f is the ADCIRC code configured for a specific problem.

5.0 CSCI DETAILED DESIGN

All routines are written in FORTRAN 77. The CSCI is platform independent and may be run on a PC or UNIX system. It can also be run on all HPC platforms.

The following paragraphs give a detailed description of the purpose, variables, logic, and constraints for the software elements in the CSCI. Description of the components of the program will be divided into design descriptions of ADCSETUP, ADCIRC, and ADCSOLVER.

5.1 CSC ADCSETUP

The program ADCSETUP completely automates the setup of ADCIRC-2DDI for any change in problem (i.e. grid, forcing, or computer). Information to modify the code is obtained from the ADCIRC grid and input files (the names are specified interactively) and from other interactive input.

5.1.1 Constraints and Limitations

None.

5.1.2 Commonly Used Input Parameters

5.1.2.1 File: adcsetup.f

Variable	Type	Description
MAXNOD	Integer	Maximum number of nodes allowable in the nearest neighbor table.

Notes: The neighbor calculation initially assumes every node is an interior node and therefore every element leg belongs to 2 elements. If this is true, each node's neighbors will be specified twice (1 time for each element that contains a particular leg.) To account for this, neighbors around a node are accumulated at only half the actual rate (i.e., every node in a particular element is assumed to have only 1 neighbor rather than 2). The problem is that boundary nodes will always have 2 neighbors that are not shared by 2 elements. Therefore, it is necessary to check which nodes are boundary nodes and add 1 to the number of neighbors for that node.

5.1.2.2 *File: fort.14*

Input variables listed by line sequence within the file:

AGRID
NE, NP
JKI, X(JKI),Y(JKI), DP(JKI) , JKI = 1,NP
JKI, NHY, NM(JKI,1), NM(JKI,2), NM(JKI,3), JKI=1,NE
NOPE
NETA
NVDLL(K), IBTYPE
NBDV(K,I) K=1,NOPE , I=1,NVDLL(K)
NBOU
NVEL
NVELL(K), IBTYPE K=1,NBOU
IF IBTYPE = 0, 1, 2, 10, 11, 12, 20, 21, 22, 30
NBVV(K,I) I=1,NVELL(K)
IF IBTYPE = 3, 13, 23
NBVV(K,I), BARLANHT(I), BARLANCFSP(I), I=1,NVELL(K)
IF IBTYPE = 4, 24
NBVV(K,I), IBCONN(I), BARINHT(I), BARINCFSB(I),BARINCFSP(I), I=1,NVELL(K)

Description of the input variables required for UNIT 14 ("fort.14").

Variable	Type	Description
AGRID	Real	Alphanumeric grid identification (<=24 characters).
BARINCFSB(I)	Real	The coefficient of free surface sub-critical flow for specified internal barrier boundary node (Typical value = 1.0).
BARINCFSP(I)	Real	The coefficient of free surface super-critical flow for specified internal barrier boundary node (Typical value = 1.0).
BARINHT(I)	Real	Internal barrier boundary node elevation above the Geoid (positive above the Geoid and negative below the Geoid). BARINHT(I) must exceed the input bathymetric values specified at the associated global node as well as at the paired node. Accounting for sign conventions for bathymetry and internal barrier height, we must satisfy BARINHT(I).GE.-DP(NBVV(K,I)) BARINHT(I).GE.-DP(IBCONN(I)) If we do not satisfy these conditions, the code will stop.
BARLANCFSP(I), I=1,NVELL(K)	Real	The coefficient of free surface supercritical flow for specified external barrier boundary node (Typical value = 1.0).

Variable	Type	Description (fort.14, Continued)
BARLANHT(I), I=1,NVELL(K)	Real	External barrier boundary node elevation above the geoid (positive above the geoid and negative below the geoid). BARLANHT(I) must exceed the input bathymetric value specified at the associated global node. Accounting for sign conventions for bathymetry and external barrier height, we must satisfy BARLANHT(I).ge.-DP(NBVV(K,I)). If we do not satisfy this condition, the code will stop.
DP(JKI), JKI=1,NP	Real	Bathymetric value. Bathymetric values are with respect to the Geoid and are positive below the Geoid and negative above the Geoid. Bathymetric values above the Geoid or any depth sufficiently small that nodes will dry, requires that the user enable the wetting/drying feature (NOLIFA=2) in the UNIT 15 input file. Nodes must be input in ascending order.
ETA2(J)	Real	If IBTYPE = 4 or 24, The known time level surface elevation solution at global node J computed in the code.
ETA2(NNBB)	Real	If IBTYPE = 3, 13 or 23, The known time level surface elevation solution at the external barrier boundary node as computed in the code.
IBCONN(I)	Integer	Paired nodes for internal barrier boundary.
IBTYPE	Integer	Boundary type (For elevation specified boundary segment K).
		= 0 Elevation specified boundary.
		= 1 Elevation specified and radiation boundary condition (not implemented).
IBTYPE	Integer	Boundary type (For normal flow segment K). (See **Appendix II Section 8.3** and **Appendix III Section 9.0** for further notes.)
		= 0 External boundary with no normal flow as an essential boundary condition and free tangential slip allowed. This represents a mainland boundary with no normal flow.
		= 1 Internal boundary with no normal flow as an essential boundary condition and free tangential slip allowed. This represents an island boundary with no normal flow.
		= 2 External boundary with specified (non-zero) normal flow as an essential boundary condition and free tangential slip allowed. This can represent a river inflow, plant discharge or open ocean boundary in which normal flow is specified.

(IBTYPE cont'd on next page)

Variable	Type	Description (fort.14, Continued)
IBTYPE (cont'd)		= 3 External boundary with a barrier which allows free surface supercritical outflow from the domain once it has been overtopped treated as an essential boundary condition and free tangential slip allowed. This represents a mainland (Weir type) barrier with normal outflow computed based on elevation at boundary node.
		= 4 Internal (island type) barrier boundary which allows free surface sub-critical and super-critical normal cross barrier inflow and out-flow once it has been overtopped. The computed cross barrier flow is treated as an essential boundary condition. Tangential slip is allowed. This boundary condition applies to (Weir type) barriers which lie within the computational domain. Normal flow across the barrier is based on elevation on both sides of the barrier.
		= 10 External boundary with no normal flow and no tangential slip as essential boundary conditions. This represents a mainland boundary with no normal flow and no slip.
		= 11 Internal boundary with no normal flow and no tangential slip as essential boundary conditions. This represents an island boundary with no normal flow and no slip.
		= 12 External boundary with specified (non-zero) normal flow and no tangential slip allowed. Both directions are treated as essential boundary conditions. This can represent a river inflow, plant discharge or open ocean boundary in which normal low is specified in addition to no slip.
		= 13 External boundary with a barrier which allows free surface supercritical outflow from the domain once it has been overtopped and no tangential slip is allowed. Both directions are treated as essential boundary conditions. This represents a mainland (Weir type) barrier with normal outflow computed based on elevation at boundary node and no slip.
		= 20 External boundary with no normal flow as a natural boundary condition and free tangential slip allowed. This represents a mainland boundary with (weak) no normal flow.
(IBTYPE cont'd on next page)		= 21 Internal boundary with no normal flow as a natural boundary condition and free tangential slip allowed. This represents an island boundary with (weak) no normal flow.

Variable	Type	Description (fort.14, Continued)
IBTYPE (cont'd)		= 22 External boundary with specified (non-zero) normal flow as a natural boundary condition and free tangential slip allowed. This can represent a river inflow, plant discharge or open ocean boundary in which (weak) normal flow is specified.
		= 23 External boundary with a barrier which allows free surface supercritical outflow from the domain once it has been overtopped as a natural boundary condition and free tangential slip allowed. This represents a mainland (Weir type) barrier with (weak) normal outflow computed based on elevation at boundary node.
		= 24 Internal (island type) barrier boundary which allows free surface subcritical and super-critical normal cross barrier inflow and out-flow once it has been overtopped. The computed cross barrier flow is treated as a natural boundary condition. Tangential slip is allowed. This boundary condition applies to (Weir type) barriers which lie within the computational domain. Normal (Weak) flow across the barrier is based on elevations on both sides of the barrier.
		= 30 In GWCE, external boundary allows the radiation of waves normal to the boundary via the flux integral. No constraint is placed on the velocity solution in the momentum equation.
JKI	Real	Index.
NBDV(K,I), K=1,NOPE, I=1,NVDLL(K)	Integer	Node numbers on elevation specified boundary segment K. The direction in which information is given does not matter Note that NVDLL(K) and IBTYPE are given first for a segment and are immediately followed by the nodes on that segment.
NBOU	Integer	Number of normal flow (or discharge) boundary segments.
NBVV(K,I), I=1,NVELL(K)	Integer	If IBTYPE = 0, 1, 2, 3, 10, 11, 12, 13, 20, 21, 22, 23, or 30 Node numbers on normal flow boundary segment K. Nodes must be specified with land always being on the right, i.e., in a counterclockwise direction for external (e.g. mainland) boundaries and a clockwise direction for internal (e.g. island) boundaries.
(cont'd on next page)		

Variable	Type	Description (fort.14, Continued)
NBVV(K,I), I=1,NVELL(K) (cont'd)		If IBTYPE = 4 or 24 Node numbers on normal flow boundary segment K which in this case is an internal barrier boundary segment. These nodes only include the front face nodes of the internal barrier. An internal barrier boundary segment consists of an island with parallel front and back faces. There is a one to one correspondence between the nodes placed on the front and back faces. Nodes are only specified for one of the two faces with the paired nodes being specified in IBCONN(I). Nodes must be specified with the inside of the barrier always being on the left, i.e., in a clockwise direction for this internal type boundary.
NE	Integer	Number of elements.
NETA	Integer	Total number of elevation specified boundary nodes.
NHY	Integer	Element type. Note that the element type is not an active variable and that only 3 node linear triangles are operational in this version of the code.
NM(JKI,1), NM(JKI,2), NM(JKI,3); JKI=1, NE	Integer	Element connectivity of node 1, node 2, and node 3 specified with a counterclockwise orientation. Elements must be read in ascending order.
NOPE	Integer	Number of elevation specified boundary forcing segments.
NP	Integer	Number of nodal points.
NVDLL(K)	Integer	Number of nodes on elevation boundary segment K.
NVEL	Integer	Total number of normal flow boundary nodes. Note that NVEL includes both front and back nodes on internal barrier type normal flow boundaries.
X(JKI),Y(JKI), JKI=1,NP	Real	X and Y coordinates. If ICS=1 in UNIT 15 then X, Y represent standard Cartesian coordinates specified in length units consistent with other UNIT 15 input (typically meters or feet). If ICS=2 in UNIT 15, then X, Y represent degrees longitude (degrees east of Greenwich is positive and degrees west of Greenwich is negative) and degrees latitude (degrees north of the equator being positive and degrees south of the equator is negative) respectively.

NOTES:

See **Appendix II, Section 8.3** for Notes on IBTYPE conditions.

5.1.2.3 File: fort.15

Input variables listed by the line sequence within the file:

RUNDES
RUNID
NFOVER
NABOUT
NSCREEN
IHOT
ICS
IM
NOLIBF
NOLIFA
NOLICA
NOLICAT
NWP
NCOR
NTIP
NWS
NRAMP
G
TAU0
DT
STATIM
REFTIM
IF (NWS.EQ.2) WTIMINC
IF (NWS.EQ.3) IREFYR, IREFMO, IREFDAY, IREFHR, IREFMIN, REFSEC
IF (NWS.EQ.3) NWLAT, NWLON, WLATMAX, WLONMIN, WLATINC, WLONINC, WTIMINC
IF (NWS.EQ.4) WTIMINC
RNDAY
DRAMP
A00, B00, C00
IF ((NOLIFA.EQ.0).OR.(NOLIFA.EQ.1)) H0
IF (NOLIFA.EQ.2) H0, NODEDRYMIN, NODEWETRMP
SLAM0, SFEA0
IF (NOLIBF.EQ.0) FFACTOR
TAU
IF (NOLIBF.EQ.1) FFACTOR
CF
IF (NOLIBF.EQ.2) FFACTOR, HBREAK, FTHETA, FGAMMA
EVM, EVC
CORI
NTIF
TIPOTAG(I)

TPK(I), AMIGT(I), ETRF(I), FFT(I), FACET(I), I=1,NTIF
NBFR
BOUNTAG(I)
AMIG(I), FF(I), FACE(I) I=1,NBFR
ALPHA
EMO(I,J), EFA(I,J) I=1,NBFR , J=1,NETA
ANGINN
NFFR
FBOUNTAG(I)
FAMIG(I), FFF(I), FFACE(I) I=1,NFFR
FALPHA
QNAM(I,J), QNPH(I,J) I=1,NFFR, J=1,NFLBN
NOUTE, TOUTSE, TOUTFE, NSPOOLE
NSTAE
XEL(I), YEL(I) I=1,NSTAE
NOUTV, TOUTSV, TOUTFV, NSPOOLV
NSTAV
XEV(I), YEV(I) I=1,NSTAV
NOUTC, TOUTSC, TOUTFC, NSPOOLC
NSTAC
XEC(I), YEC(I) I=1,NSTAC
NOUTGE, TOUTSGE, TOUTFGE, NSPOOLGE
NOUTGV, TOUTSGV, TOUTFGV, NSPOOLGV
NOUTGC, TOUTSGC, TOUTFGC, NSPOOLGC
NOUTGW, TOUTSGW, TOUTFGW, NSPOOLGW
NHARFR
HAFNAM(I)
HAFREQ(I), HAFF(I), HAFACE(I) I=1,NHARFR
THAS, THAF, NHAINC, FMV
NHASE, NHASV, NHAGE, NHAGV
NHSTAR, NHSINC
ITITER, ISLDIA, CONVCR, ITMAX

Description of input variables required for UNIT 15 ("fort.15").

Variable	Type	Description
A00, B00, C00	Real	Time weighting factors (K+1, K, K-1) in wave equation.
ALPHA	Real	Alphanumeric descriptor of the constituent name. See description of EMO(I,J), EFA(I,J) (<= 10 characters).
AMIG(I), FF(I), FACE(I), I=1,NBFR	Real Real Real	Forcing frequency, nodal factor, equilibrium argument in degrees for tidal forcing on elevation specified boundaries. These values are preceded by BOUNTAG(I).

Variable	Type	Description (fort.15, Continued)
ANGINN	Real	Flow boundary nodes which are set up to have a normal flow essential boundary condition and have an inner angle less than ANGINN (specified in degrees) will have the tangential velocity zeroed. In either case, the normal velocity will be determined from the essential boundary condition.
BOUNTAG(I)	Real	An alphanumeric descriptor of the constituent name. See description of AMIG(I),FF(I),FACE(I).
CONVCR	Real	Absolute convergence criteria (should be no smaller than 500 times the machine precision) (See Notes in **Appendix II, Section 8.4**).
CORI	Real	Constant Coriolis coefficient. This value is always read in. However, it is only used in the computations when NCOR=0.
DRAMP	Real	Value in days used in computing the ramp function which is evaluated as RAMP=TANH (2.0*IT*DT/(24.0*3600.0*DRAMP)) where it equals the time step index. Note that DRAMP is approximately equal to the number of days at which RAMP=0.96.
DT	Real	Time step (in seconds).
EMO(I,J), EFA(I,J), I=1,NBFR , J=1,NETA	Real Real	Amplitude and phase (in degrees) of the harmonic forcing function at the elevation specified boundaries for frequency I and elevation specified boundary forcing node J. Note that the parameter NETA is defined and read in from UNIT 14 input. The forcing values are preceded by an alphanumeric descriptor ALPHA to facilitate verifying that the correct data matches a given frequency.
EVC	Real	Lateral diffusivity for transport equation. Units of L^2/T. Note: it is only necessary to specify EVC if IM=10, otherwise, it can be omitted.
EVM	Real	Lateral viscosity for momentum equations. Units of L^2/T.
FALPHA	Real	Alphanumeric descriptor of the constituent name. See description of QNAM (I, J), QNPH (I, J) (<=10 characters).
FAMIG(I), FFF(I), FFACE(I), I=1,NFFR	Real Real Real	Forcing frequency, nodal factor, equilibrium argument in degrees for periodic normal flow forcing on flow boundaries. These values are preceded by FBOUNTAG(I).
FBOUNTAG(I)	Real	An alphanumeric descriptor of the constituent name. See description of FAMIG(I), FFF(I), FFACE(I).

Variable	Type	Description (fort.15, Continued)
FFACTOR (if NOLIBF=0)	Real	If NOLIBF=0, Bottom friction coefficient. The standard linear bottom friction coefficient TAU = FFACTOR. Note, in this case FFACTOR should be equal to the specified TAU0. FFACTOR is always read in, however, it is only used in the computations when NWP=0.
		If NOLIBF = 1, Bottom friction coefficient. The standard quadratic bottom friction coefficient CF = FFACTOR. FFACTOR is always read in, however, it is only used in the computations when NWP=0.
		If NOLIBF = 2, Equals CFMIN. This is the standard quadratic friction coefficient that is approached in deep water.
FGAMMA	Real	A parameter that determines how the friction factor increases as the water depth decreases. Setting this to 1/3 gives a Manning friction law type of behavior (FGAMMA=1/3 is recommended).
FMV	Real	Fraction of the harmonic analysis period (extending back from the end of the harmonic analysis period) to use for comparing the water elevation and velocity means and variances from the raw model time series with corresponding means and variances of a time series resynthesized from the harmonic constituents. This comparison is helpful for identifying numerical instabilities and for determining how complete the harmonic analysis was (See **Appendix II, Section 8.2** for further notes).
		= 0 Do not compute any means and variances.
		= 0.1 Compute means and variances over final 10% of period used in harmonic analysis.
		= 1.0 Compute means and variances over entire period used in harmonic analysis.
FTHETA	Real	A parameter that determines how rapidly the hybrid relationship approaches each asymptotic limit. (FTHETA=10 is recommended).
G	Real	Gravitational constant. Note that the code always operates in seconds and that the specified time units for G must be seconds. When ICS = 1 and when either NTIP and or NCOR = 1, G must be specified equal to 9.81 m/sec*sec since the program only operates in metric units for this case. When ICS = 2, G must be specified equal to 9.81 m/sec*sec since the program only operates in metric units for this case.

Variable	Type	Description (fort.15, Continued)
H0 (if NOLIFA=0 or NOLIFA=1)	Real	Minimum bathymetric depth. All bathymetric depths in the input grid (UNIT 14) are checked against this depth and set equal to H0 if they are initially less than H0.
H0 (if NOLIFA=2)	Real	Nominal minimum water depth for drying (Typical value = 0.01 M).
HAFNAM(I)	Real	Alphanumeric descriptor of the constituent name (<=10 characters). See description of HAFF(I), HAFREQ(I), HAFACE(I).
HAFREQ(I), HAFF(I), HAFACE(I), I=1,NHARFR	Real	Frequency (rad/s). Nodal factor. Equilibrium argument (degrees). Note: This value is preceded by HAFNAM, an alphanumeric descriptor. If a steady component will be included in the harmonic analysis, this must be the first constituent listed (i.e., the constituent corresponding to I=1).
HBREAK	Real	The break depth. In deeper water the friction relation looks like a standard quadratic relation. In shallower water the friction factor increases as the depth decreases (e.g. like a Manning type friction law). (HBREAK = 1 m)
ICS	Integer	Parameter that determines in which coordinate system the calculations are performed.
		= 1 Standard Cartesian coordinate form of the governing equations is used in the model. Input coordinates are in standard length units (as determined by G). If NTIP = 1 and/or NCOR = 1, then an inverse Carte Parallelogrammatique Projection is used in order to obtain the coordinates for the nodes in degrees longitude and latitude so that the variable Coriolis and tidal potential forcing function can be computed. However, if the Coriolis parameter varies significantly over the domain and/or the domain is large enough that tidal potential. forcing is significant, it is recommended to operate the code with ICS =2 (with the governing equations based on spherical equations).
		= 2 Governing equations are based on the spherical equations transformed into using a Carte Parallelogrammatique Projection (CPP). The input coordinates are in degrees longitude and latitude that are then transformed using the CPP.
IHOT (cont'd on next page)	Integer	Parameter that controls whether the model is hot started. Note: The hot start facility is only available for 2DDI runs.
		= 0 Do not hot start the model (use cold start).

Variable	Type	Description (fort.15, Continued)
IHOT (cont'd)		= 67 Hot start model using input read on UNIT 67 (fort.67).
		= 68 Hot start model using input read on UNIT 68 (fort.68).
IM	Integer	Model type.
		= 0 2DDI model, hydrodynamics only.
		= 1 3D VS model, hydrodynamics only.
		= 2 3D DSS model, hydrodynamics only.
		= 10 2DDI model, hydrodynamics and transport.
IREFDAY	Integer	Day of the start of the simulation. Must be consistent with the times in the Fleet Numeric wind file.
IREFHR	Integer	Hour of the start of the simulation. Must be consistent with the times in the Fleet Numeric wind file.
IREFMIN	Integer	Minute of the start of the simulation. Must be consistent with the times in the Fleet Numeric wind file.
IREFMO	Integer	Month of the start of the simulation. Must be consistent with the times in the Fleet Numeric wind file.
IREFYR	Integer	Year of the start of the simulation. Must be consistent with the times in the Fleet Numeric wind file.
ISLDIA	Integer	Parameter that controls the level of diagnostic output from the ITPACK 2D solver (See Notes in **Appendix II, Section 8.4**).
		= 0 Fatal error messages only from ITPACKV 2D (UNIT 33).
		= 1 Warning messages and minimum output from ITPACKV 2D (UNIT 33).
		= 2 Reasonable summary of algorithm progress from ITPACKV 2D (UNIT 33)..
		= 3 Parameter values and informative comments from ITPACKV 2D (UNIT 33).
		= 4 Approximate solution after each iteration from ITPACKV 2D (UNIT 33).
		= 5 Original system from ITPACKV 2D (UNIT 33).
ITITER	Integer	Parameter that specifies the solver that will be used for GWCE (See Notes in **Appendix II, Section 8.4**)..
		= -1 Only for lumped, explicit GWCE, matrix is diagonal and no external solver is needed.
		= 0 Use direct banded solver.
		= 1 Use iterative JCG solver (from ITPACKV 2D).
		= 2 Use iterative JSI solver (from ITPACKV 2D).
		= 3 Use iterative SOR solver (from ITPACKV 2D).
(cont'd on next page)		= 4 Use iterative SSORCG solver (from ITPACKV 2D).
		= 5 Use iterative SSORSI solver (from ITPACKV 2D).

Variable	Type	Description (fort.15, Continued)
ITITER (cont'd)		= 6 Use iterative RSCG solver (from ITPACKV 2D).
		= 7 Use iterative RSSI solver (from ITPACKV 2D).
ITMAX	Integer	Maximum number of iterations each time step (See Notes in **Appendix II, Section 8.4**).
NABOUT	Integer	Parameter for abbreviated output on UNIT 16. If NABOUT=1, then the output to UNIT 16 will be abbreviated by not echo printing most of the UNIT 14, 21, and 22 input information. If NABOUT is not specified equal to 1, all the information from all input files will be echo printed to UNIT 16.
NBFR	Integer	Number of periodic forcing frequencies on elevation specified boundaries. If NBFR=0 and a nonzero number of elevation specified boundary segments are included in the UNIT 14 input, the elevation boundary condition is assumed to be non-periodic and will be read in from UNIT 19. For reasons of backward compatibility, NBFR is included in the UNIT 15 file regardless of whether any elevation specified boundaries (IBTYPE=0) are defined in the UNIT 14 input.
NCOR	Integer	Coriolis option parameter.
		= 0 For spatially constant Coriolis parameter.
		= 1 For spatially variable Coriolis parameter.
NFFR	Integer	Number of frequencies in the specified normal flow external boundary condition. If NFFR=0, the normal flow boundary condition is assumed to be non-periodic and will be read in from UNIT 20. NFFR is only included in the UNIT 15 file if one or more specified (non-zero) normal flow external boundaries were defined in the UNIT 14 input, (IBTYPE = 2,12 or 22).
NFOVER	Integer	Non-fatal error override option. When NFOVER=1, all checked input parameters which are inconsistent will in most cases be automatically corrected to a default or corrected value and execution will continue. Be sure to read the nonfatal warning messages to see what the program has done. If NFOVER≠1, all checked input parameters which are inconsistent will lead to program termination. Note that NFOVER does not effect all fatal warning checks, which will always stop execution. Furthermore all non-fatal runtime error checks are treated in the same manner as input parameter checks.
NHAGE	Integer	Parameter that controls the spatial locations where harmonic analysis is performed.
(cont'd on next page)		= 0 No harmonic analysis is performed for global elevations.

Variable	Type	Description (fort.15, Continued)
NHAGE (cont'd)		= 1 Harmonic analysis is performed for global elevations (output on UNIT 53).
NHAGV	Integer	Parameter that controls the spatial locations where harmonic analysis is performed.
		= 0 No harmonic analysis is performed for global velocities.
		= 1 Harmonic analysis is performed for global velocities (output on UNIT 54).
NHAINC	Integer	The number of time steps at which information is harmonically analyzed, i.e. information every NHAINC time steps after THAS is used in harmonic analysis.
NHARFR	Integer	Number of frequencies to include in harmonic analysis of model results. Note: Harmonic output is only available for 2DDI elevation and velocity.
NHASE	Integer	Parameter that controls the spatial locations where harmonic analysis is performed. NOTE: The stations are as specified in the section on time series station velocity output.
		= 0 No harmonic analysis is performed at the selected elevation recording stations.
		= 1 Harmonic analysis is performed at the selected elevation recording stations (output on UNIT 51).
NHASV	Integer	Parameter that controls the spatial locations where harmonic analysis is performed. NOTE: The stations are as specified in the section on time series station velocity output.
		= 0 No harmonic analysis is performed at the selected velocity recording stations.
		= 1 Harmonic analysis is performed at the selected velocity recording stations (output on UNIT 52).
NHSINC	Integer	The number of time steps at which hot start output file is generated (Hot start file is generated every NHSINC time steps).
NHSTAR	Integer	Parameter that controls the generation of hot start output.
		= 0 No hot start output files generated.
		= 1 Hot start output files generated.
NODEDRYMIN	Integer	Minimum number of time steps a node must remain dry after drying before it can wet (Typical value = 5-20).
NODEWETRMP	Integer	Number of time steps after a node wets over which the minimum water depth for drying is ramped up from 0 to H0. This is done so that small oscillations in the water depth after a node wets do not cause the node to dry again (Typical value = 5-20).

Variable	Type	Description (fort.15, Continued)
NOLIBF	Integer	A model option parameter which determines whether bottom stress is applied as a linear relationship or using a nonlinear parameterization.
		= 0 Linear bottom friction. Friction coefficient defined under FFACTOR (NOLIBF=0).
		= 1 Nonlinear bottom friction - standard quadratic bottom friction law. Friction coefficient defined under FFACTOR (NOLIBF=0).
		= 2 Nonlinear bottom friction - Hybrid between friction law in which the friction coefficient increases as the depth decreases (e.g. a Manning friction law) at shallow depths and a standard quadratic friction law at deep depths. CF = CFMIN*[1+(HBREAK/H)**FTHETA]**(FGAMMA/FTHETA) The required parameters are specified under FFACTOR (NOLIBF=2), HBREAK, FTHETA, and FGAMMA.
NOLICA	Integer	Model option parameter that determines whether the advective terms (with the exception of the time derivative portion that shows up in the GWCE) are turned on.
		= 0 Advective terms containing spatial derivatives are not included in the computations.
		= 1 Advective terms containing spatial derivatives are included in the computations. Note, when the spatial derivative portions of the advective terms are included, the time derivative portion of the advective terms (in the GWCE) should also be included (i.e. when NOLICA=1, NOLICAT=1).
NOLICAT	Integer	Model option parameter that determines whether the time derivative component of the advective terms that show up in the GWCE are included in the computations.
		= 0 The time derivative components of the advective terms in the GWCE are not included in the computations.
		= 1 The time derivative components of the advective terms in the GWCE are included in the computations. Note: These terms should be turned on if either the finite amplitude or the spatial derivative components of the advective terms are turned on for proper mass conservation and solution consistency.

Variable	Type	Description (fort.15, Continued)
NOLIFA	Integer	A model option parameter that determines how the finite amplitude component of the total depth is considered. Note, whether wetting and drying is enabled changes the meaning of H0 and the need to specify several additional parameters on this line. Note, when the finite amplitude terms are turned on the time derivative portion of the advective terms should also be turned on for proper mass conservation and consistency (i.e. when NOLIFA > 0, NOLICAT should be set equal to 1)
		= 0 Finite amplitude terms in the equations are not turned on, wetting and drying of elements is disabled.
		= 1 Finite amplitude terms in the equations are turned on, wetting and drying of elements is disabled.
		= 2 Finite amplitude terms in the equations are turned on, wetting and drying of elements is enabled.
NOUTC	Integer	Output parameter that controls the time series output provided for concentration solutions at selected concentration recording stations (UNIT 71 output).
		= -2 Output is provided at the selected concentration recording stations in binary format. Following a hot start, a new UNIT 71 file is created.
		= -1 Output is provided at the selected concentration recording stations in standard ASCII format. Following a hot start, a new UNIT 71 file is created.
		= 0 No output is provided at the selected concentration recording stations.
		= 1 Output is provided at the selected concentration recording stations in standard ASCII format. Following a hot start, continued output is merged into the existing UNIT 71 file.
		= 2 Output is provided at the selected concentration recording stations in binary format. Following a hot start, continued output is merged into the existing UNIT 71 file.
NOUTE	Integer	Output parameter that controls the time series output provided for elevation solutions at selected elevation recording stations (UNIT 61 output).
		= -2 Output is provided at the selected elevation recording stations in binary format. Following a hot start, a new UNIT 61 file is created.
(cont'd on next page)		= -1 Output is provided at the selected elevation recording stations in standard ASCII format. Following a hot start, a new UNIT 61 file is created.

Variable	Type	Description (fort.15, Continued)
NOUTE (cont'd)		= 0 No output is provided at the selected elevation recording stations.
		= 1 Output is provided at the selected elevation recording stations in standard ASCII format. Following a hot start, continued output is merged into the existing UNIT 61 file.
		= 2 Output is provided at the selected elevation recording stations in binary format. Following a hot start, continued output is merged into the existing UNIT 61 file.
NOUTGC	Integer	Output parameter that controls the time series output provided for global concentration solutions at all nodes within the domain (UNIT 73 output).
		= -2 Global concentration output is provided in binary format. Following a hot start, a new UNIT 73 file is created.
		= -1 Global concentration output is provided in standard ASCII format. Following a hot start, a new UNIT 73 file is created.
		= 0 No global concentration output is provided.
		= 1 Global concentration output is provided in standard ASCII format. Following a hot start, continued output is merged into the existing UNIT 73 file.
		= 2 Global concentration output is provided in binary format. Following a hot start, continued output is merged into the existing UNIT 73 file.
NOUTGE	Integer	Output parameter that controls the time series output provided for global elevation solutions at all nodes within the domain (UNIT 63 output).
		= -2 Global elevation output is provided in binary format. Following a hot start, a new UNIT 63 file is created.
		= -1 Global elevation output is provided in standard ASCII format. Following a hot start, a new UNIT 63 file is created.
		= 0 No global elevation output is provided.
		= 1 Global elevation output is provided in standard ASCII format. Following a hot start, continued output is merged into the existing UNIT 63 file.
		= 2 Global elevation output is provided in binary format. Following a hot start, continued output is merged into the existing UNIT 63 file.

Variable	Type	Description (fort.15, Continued)
NOUTGV	Integer	Output parameter that controls the time series output provided for global velocity solutions at all nodes within the domain (UNIT 64 output).
		= -2 Global velocity output is provided in binary format. Following a hot start, a new UNIT 64 file is created.
		= -1 Global velocity output is provided in standard ASCII format. Following a hot start, a new UNIT 64 file is created.
		= 0 No global velocity output is provided.
		= 1 Global velocity output is provided in standard ASCII format. Following a hot start, continued output is merged into the existing UNIT 64 file.
		= 2 Global velocity output is provided in binary format. Following a hot start, continued output is merged into the existing UNIT 64 file.
NOUTGW	Integer	Output parameter that controls the time series output provided for wind stress at all nodes within the domain (UNIT 74 output). Note: This line is only read in if wind forcing is included in the model run (i.e. NWS>0).
		= -2 Global wind stress output is provided in binary format. Following a hot start, a new UNIT 74 file is created.
		= -1 Global wind stress output is provided in standard ASCII format. Following a hot start, a new UNIT 74 file is created.
		= 0 No global wind stress output is provided.
		= 1 Global wind stress output is provided in standard ASCII format. Following a hot start, continued output is merged into the existing UNIT 74 file.
		= 2 Global wind stress output is provided in binary format. Following a hot start, continued output is merged into the existing UNIT 74 file.
NOUTV	Integer	Output parameter that controls the time series output provided for velocity solutions at selected velocity recording stations (UNIT 62 output).
		= -2 Output is provided at the selected velocity recording stations in binary format. Following a hot start, a new UNIT 62 file is created.
		= -1 Output is provided at the selected velocity recording stations in standard ASCII format. Following a hot start, a new UNIT 62 file is created.
(cont'd on next page)		= 0 No output is provided at the selected velocity recording stations.

Variable	Type	Description (fort.15, Continued)
NOUTV (cont'd)		= 1 Output is provided at the selected velocity recording stations in standard ASCII format. Following a hot start, continued output is merged into the existing UNIT 62 file.
		= 2 Output is provided at the selected velocity recording stations in binary format. Following a hot start, continued output is merged into the existing UNIT 62 file.
NRAMP	Integer	Ramp option parameter.
		= 0 No ramp function is used with forcing functions.
		= 1 A hyperbolic tangent ramp function is specified and applied to the surface elevation specified boundary conditions, the tidal potential forcing function, the wind and atmospheric pressure forcing functions and the flux input boundary condition.
NSCREEN	Integer	Parameter which controls output to UNIT 6.
		= 0 No output is directed to UNIT 6.
		= 1 Screen output is directed to UNIT 6.
NSPOOLC	Integer	The number of time steps at which information is spooled to UNIT 71, i.e. the output is spooled to UNIT 71 every NSPOOLC time steps after TOUTSC. Note: This line is only read in if transport is included in the model run (i.e. IM=10).
NSPOOLE	Integer	The number of time steps at which information is spooled to UNIT 61, i.e. the output is spooled to UNIT 61 every NSPOOLE time steps after TOUTSE.
NSPOOLGC	Integer	The number of time steps at which information is spooled to UNIT 73, i.e. the output is spooled to UNIT 73 every NSPOOLGC time steps after TOUTFGC. Note: This line is only read in if transport is included in the model run (i.e. IM=10).
NSPOOLGE	Integer	The number of time steps at which information is spooled to UNIT 63, i.e. the output is spooled to UNIT 63 every NSPOOLGE time steps after TOUTFGE.
NSPOOLGV	Integer	The number of time steps at which information is spooled to UNIT 64, i.e. the output is spooled to UNIT 64 every NSPOOLGV time steps after TOUTSGV.
NSPOOLGW	Integer	The number of time steps at which information is spooled to UNIT 74, i.e. the output is spooled to UNIT 74 every NSPOOLGW time steps after TOUTSGW. Note: This line is only read in if wind forcing is included in the model run (i.e. NWS>0).

Variable	Type	Description (fort.15, Continued)
NSPOOLV	Integer	The number of time steps at which information is spooled to UNIT 62, i.e. the output is spooled to UNIT 62 every NSPOOLV time steps after TOUTSV.
NSTAC	Integer	The number of concentration recording stations. Note this line is only read in if transport is included in the model run (i.e. IM=10). This is read in even if NOUTC=0.
NSTAE	Integer	The number of elevation recording stations (This is always read in regardless of the value of NOUTE).
NSTAV	Integer	The number of velocity recording stations (this is always read in regardless of the value of NOUTV).
NTIF	Integer	Number of tidal potential constituents.
NTIP	Integer	Tidal potential and self attraction/load tide option parameter.
		= 0 When the tidal potential and self attraction/load tide functions are not used.
		= 1 When only tidal potential function is used.
		= 2 When both tidal potential and self attraction/load tide functions are used. In this case the self-attraction/ load tide information is read in for each constituent at each node in the grid on UNIT 23.
NWLAT	Integer	Number of latitude values in Fleet Numeric wind file.
NWLON	Integer	Number of longitude values in Fleet Numeric wind file.
NWP	Integer	Bottom friction option parameter.
		= 0 For spatially constant linear or quadratic bottom friction coefficient.
		= 1 For spatially varying linear or quadratic bottom friction. UNIT 21 must be opened; Will not work for NOLIBF = 2.
NWS	Integer	Wind stress and surface pressure option parameter (See **Appendix II Section 8.1** for further notes).
		= 0 No wind stress or surface pressure is applied.
		> 0 Spatially varying wind stress and surface pressure are applied.
		= 1 Wind stress and pressure are read in at all grid nodes every time step from UNIT 22.
		= 2 Wind stress and pressure are read from the UNIT 22 file at all ADCIRC grid nodes at a wind time interval that does not equal the model time step. Interpolation in time is used to synchronize the wind and pressure information with the model time step. The wind time interval must be specified later in the UNIT 15 file.

(cont'd on next page)

Variable	Type	Description (fort.15, Continued)
NWS (cont'd)		= 3 Wind velocity is read in from an outdated U.S. Navy Fleet Numeric format wind file on UNIT 22. This data is interpolated in space onto the ADCIRC grid. Interpolation in time is used to synchronize the wind information with the model time step. Garret's formula is used to compute wind stress from velocity. Several parameters describing the U.S. Navy Fleet Numeric wind file must be specified later in the UNIT 15 file.
		= 4 Wind velocity and pressure are read in at selected ADCIRC grid nodes from a PBL/JAG format file on UNIT 22. Interpolation in time is used to synchronize the wind information with the model time step. Garret's formula is used to compute wind stress from velocity. The wind time interval must be specified later in the UNIT 15 file.
		= 10 10 m high wind velocity and surface pressure are read in from NWS AVN model MET files. These files are in binary and have been created from a larger GRIB format file using UNPKGRB1. Each file is assumed to contain data on a Gaussian longitude/latitude grid at a single time. Data consecutive files are assumed to be separated by 6 hours in time (See **Appendix II, Section 8.1** for further notes).
		=11 10 m high wind velocity and surface pressure are read in from stripped down NWS ETA-29 MET files. These files are in binary (See **Appendix II, Section 8.1** for further notes).
QNAM(I,J), QNPH(I,J), I=1,NFFR, J=1,NFLBN	Real	Amplitude and phase (in degrees) of the periodic normal flow/unit width (e.g. m^2/s) for frequency I and "specified normal flow" boundary node J. A positive flow/unit width is into the domain and a negative flow/unit width is out of the domain. Note: NFLBN is the total number of specified normal flow external boundary nodes that were designated in the UNIT 14 file as having a non zero normal flow (IBTYPE=2,12,22). The forcing values are preceded by an alphanumeric descriptor FALPHA to facilitate verifying that the correct data matches a given frequency.
REFSEC	Real	Second of the start of the simulation. Must be consistent with the times in the Fleet Numeric wind file.

Variable	Type	Description (fort.15, Continued)
REFTIM	Real	Reference time (in days) with respect to which all nodal factors and equilibrium arguments are computed (i.e. FFT(I), FACET(I), FF(I), FACE(I), FFHA(I), FACEHA(I)).
RNDAY	Real	Total number of days of numerical simulation.
RUNDES	Real	Alphanumeric run description (<= 32 characters).
RUNID	Real	Alphanumeric run identification (<= 24 characters).
SLAM0, SFEA0	Real	Longitude and latitude on which CPP projection is centered (in degrees).
STATIM	Real	Starting simulation time (in days). Defines the starting time for the simulation. The first time step computes results for TIME=STATIM+DT.
TAU0	Real	Generalized wave-continuity equation weighting factor. TAUO weighs the primitive and wave portions of the GWCE.
		= 0 Pure wave equation.
		>> 1 Primitive continuity equation.
THAF	Real	The number of days after which data ceases to be harmonically analyzed (THAF is relative to STATIM).
THAS	Real	The number of days after which data starts to be harmonically analyzed (THAS is relative to STATIM).
TIPOTAG(I)	Real	Alphanumeric descriptor of the constituent name. See description of TPK(I), AMIGT(I), ETRF(I), FFT(I) AND FACET(I).
TOUTFC	Real	The number of days after which concentration station data ceases to be recorded to UNIT 71 (TOUTFC is relative to STATIM).
TOUTFE	Real	The number of days after which elevation station data ceases to be recorded to UNIT 61 (TOUTFE is relative to STATIM).
TOUTFGC	Real	The number of days after which global concentration data ceases to be recorded to UNIT 73 (TOUTFGC is relative to STATIM).
TOUTFGE	Real	The number of days after which global elevation data ceases to be recorded to UNIT 63 (TOUTFGE is relative to STATIM).
TOUTFGV	Real	The number of days after which global velocity data ceases to be recorded to UNIT 64 (TOUTFGV is relative to STATIM).

Variable	Type	Description (fort.15, Continued)
TOUTFGW	Real	The number of days after which global wind stress data ceases to be recorded to UNIT 74 (TOUTFGW is relative to STATIM). Note: This line is only read in if wind forcing is included in the model run (i.e. NWS>0).
TOUTFV	Real	The number of days after which velocity station data ceases to be recorded to UNIT 62 (TOUTFV is relative to STATIM).
TOUTSC	Real	The number of days after which concentration station data is recorded to UNIT 71 (TOUTSC is relative to STATIM).
TOUTSE	Real	The number of days after which elevation station data is recorded to UNIT 61 (TOUTSE is relative to STATIM).
TOUTSGC	Real	The number of days after which global concentration data is recorded to UNIT 73 (TOUTSGC is relative to STATIM).
TOUTSGE	Real	The number of days after which global elevation data is recorded to UNIT 63 (TOUTSGE is relative to STATIM).
TOUTSGV	Real	The number of days after which global velocity data is recorded to UNIT 64 (TOUTSGV is relative to STATIM).
TOUTSGW	Real	The number of days after which global wind stress data is recorded to UNIT 74 (TOUTSGW is relative to STATIM). Note: This line is only read in if wind forcing is included in the model run (i.e. NWS>0).
TOUTSV	Real	The number of days after which velocity station data is recorded to UNIT 62 (TOUTSV is relative to STATIM).
TPK(I), AMIGT(I), ETRF(I), FFT(I), FACET(I); I=1,NTIF	Real Real Real Real Real	Tidal potential amplitude, frequency, earth tide potential reduction factor (generally taken to be 0.690 for all constituents (Hendershott) but for more precise calculations can take on slightly different values (e.g. see Wahr, 1981)), nodal factor, and equilibrium argument in degrees. These values are preceded by TIPOTAG(I).
VELMIN	Real	Minimum velocity for wetting. A node along the wet/dry interface wets if the velocity components at that node that are directed toward all neighboring dry nodes exceed VELMIN (Typical value = 0.05 m/s).
WLATINC	Real	Latitude increment (deg) of Fleet Numeric wind data in the UNIT 22 file (must be > 0).
WLATMAX	Real	Maximum latitude (deg) of Fleet Numeric wind data in the UNIT 22 file (< 0 south of the equator).
WLONINC	Real	Longitude increment (deg) of Fleet Numeric wind data in the UNIT 22 file (must be > 0).
WLONMIN	Real	Minimum longitude (deg) of Fleet Numeric wind data in the UNIT 22 file (< 0 west of Grenich meridian).

Variable	Type	Description (fort.15, Continued)
WREFTIM	Real	The start time of the simulation in seconds since the beginning of the calendar year (cannot combine data from two different years).
WTIMINC	Real	If NWS = 2 or 4, Time interval (in seconds) between Fleet Numeric wind and pressure data in the UNIT 22 file.
		If NWS = 3, Time interval (in seconds) between Fleet Numeric wind data in the UNIT 22 file.
XEC(I), YEC(I) I=1,NSTAC	Real Real	The coordinates of the concentration recording station I, for all NSTAC stations. Note: This line is only read in if transport is included in the model run (i.e. IM=10) Note: The coordinates must be consistent (i.e. Cartesian or spherical) with the UNIT 14 file and the coordinate designation parameter, ICS, in the UNIT 15 file. Note: If a concentration recording station is input which does not lie within the computational domain, a non-fatal error message will appear. If NFOVER has been set equal to 1, the code will estimate the nearest element and use that as the basis of interpolation. A proximity index is printed out in the UNIT 16 file that indicates how close or far the station coordinates are from the nearest element. This index may be interpreted as the number of elements that the station lies from the nearest element.
XEL(I), YEL(I) I=1,NSTAE	Real Real	The coordinates of the elevation recording station I, for all NSTAE stations. If ICS = 1, coordinates are input as standard Cartesian. If ICS = 2, coordinates are input as degrees longitude and latitude. If an elevation recording station is input which does not lie within the computational domain, a non-fatal error message will appear. If NFOVER has been set equal to 1, the code will estimate the nearest element and use that as the basis of interpolation. A proximity index is also printed out that indicates how close or far the station coordinates are from the nearest element. This index may be interpreted as the number of elements that the station lies from the nearest element.
XEV(I),YEV(I) I=1,NSTAV	Real	The coordinates of the velocity recording station I, for all NSTAV stations. If ICS = 1, coordinates are input as standard Cartesian. If ICS = 2, coordinates are input as degrees longitude and latitude. If a velocity recording station is input which does not lie within the computa-tional domain, a non-fatal error message will appear. If NFOVER has been set equal to 1, the code will estimate the nearest element and use that as the basis of inter-polation. A proximity index is also printed out that indicates how close or far the station coordinates are from the nearest element. This index may be interpreted as the number of elements that the station lies from the nearest element.

5.2 CSC ADCIRC

ADCIRC-2DDI is a finite element, hydrodynamic model that applies tidal and meteorological forcing to produce a depth-integrated velocity field and water elevations. The model is based on the mass and momentum conservation equations, which have been formulated using the traditional hydrostatic pressure and Boussinesq approximations. They have been discretized in space and time using the finite element (FE) and the finite difference (FD) methods, respectively. First, elevation is obtained from the solution of the depth-integrated continuity equation in Generalized Wave-Continuity Equation (GWCE) form. Then, velocity is obtained from a solution of the momentum equations. The ADCIRC code consists of 40 subroutines and 30 common blocks.

The GWCE is solved using either a consistent or a lumped mass matrix as specified through adcsetup.f, and an implicit or explicit time stepping scheme defined by the variable time weighting coefficients specified in UNIT 15. If a lumped, fully explicit formulation is specified, no matrix solver is necessary. In all other cases the GWCE is best solved using the Jacobi preconditioned iterative solver from the ITPACKV 2D package.

The momentum equations are lumped generating a diagonal matrix system that requires no matrix solver.

5.2.1 Constraints and Limitations

The ADCIRC-2DDI code is modular in that switches for various dynamical terms/forcings are set to on or off in the fort.15 file. ADCIRC-2DDI is written in FORTRAN 77.

5.2.2 Initialization

The ADCIRC-2DDI model is initialized by running the program adcsetup.f. Information used to dimension and configure the code is obtained from the ADCIRC grid (fort.14) and parameter input file (fort.15). The names of these files as well as additional information specifying computer platform and compiler type are interactively input.

5.2.3 Boundary Conditions

ADCIRC boundary conditions include:

- specified elevation (harmonic tidal constituents or time series)
- specified normal flow (harmonic tidal constituents or time series)
- zero normal flow
- slip or no slip boundary layer conditions for velocity
- external barrier overflow out of the domain
- internal barrier overflow between sections of the domain
- spatial and temporal surface stress (wind and/or wave radiation stress)
- spatial and temporal atmospheric pressure
- outward radiation of waves in GWCE via a flux integral (Sommerfeld condition)

5.2.4 *Logic and Basic Equations*

For a complete description of the governing equations, numerical formulation, and implementation of ADCIRC-2DDI, the user is referred to Luettich, Westerink, and Scheffner (1992). ADCIRC-2DDI User's Manual also offers information about governing equations and is found on the web at:
http://www.marine.unc.edu/C_CATS/adcirc/document/ADCIRC_main_frame.html.

5.2.5 *Commonly Used Input Parameters*

5.2.5.1 *Dimension parameters*

The following table is a list and description of the parameters that must be set within the main code and the subroutines to control the dimensioning of arrays.

Variable	Type	Description
MNP	Integer	Maximum number of nodal points.
MNWP	Integer	1 if no meteorological forcing, = MNP if meteorological forcing.
MNE	Integer	Maximum number of elements.
MNBW	Integer	Maximum half band width of wave equation matrix (if an iterative solver is used, MNBW is not used for anything, therefore it can be set to any value or removed from the parameter list).
MNBFR	Integer	Maximum number of periodic specified elevation boundary forcing constituents.
MNFFR	Integer	Maximum number of periodic specified normal flow boundary forcing constituents.
MNTIF	Integer	Maximum number of tidal potential constituents.
MNBOU	Integer	Maximum number of flow boundary segments including external and internal segments.
MNVEL	Integer	Maximum total number of flow boundary nodes +1.
MNOPE	Integer	Maximum number of elevation boundary segments.
MNETA	Integer	Maximum total number of specified elevation boundary nodes.
MNSTAE	Integer	Maximum number of elevation recording stations.
MNSTAV	Integer	Maximum number of velocity recording stations.
MNSTAC	Integer	Maximum number of concentration recording stations.
MNHARF	Integer	Maximum number of constituents to include in harmonic analysis of model results (Note: To run on the CRAY, if 4*(MNHARF/4)<MNHARF, MNHARF should be rounded up to the next larger integer that is evenly divisible by 4).
MNEI	Integer	1 + maximum number of nodes connected to any one node in the finite element grid.

5.2.5.2 File: fort.14

See description of variables in Section 5.1.2.2.

5.2.5.3 File: fort.15

See description of variables in Section 5.1.2.3.

5.2.5.4 File: fort.19

Input variables listed by the line sequence within the file:

ETIMINC
NETA
ESBIN

Description of transient elevation data read in for elevation specified boundaries from UNIT 19.

Variable	Type	Description
ESBIN(I)	Real	Elevation (referenced to the GEOID) at specified elevation node I. The sequencing is assumed to match what is defined in the elevation specified boundary condition part of the UNIT 14 file.
ETIMINC	Real	Time increment (secs) between consecutive sets of elevation specified boundary condition values contained in this file.
NETA	Integer	The total number of elevation specified boundary nodes.

NOTES:
- This file is only used if NBFR=0 (UNIT 15 file) and NOPE>0 (UNIT 14 file).
- The first set of elevation values are provided at TIME=STATIM. Subsequent sets of elevation values are provided every ETIMINC.
- Enough sets of elevation values must be provided to extend for the entire model run, otherwise the run will crash!

5.2.5.5 *File: fort.20*

Input variables listed by the line sequence within the file:

FTIMINC
NFLBN
QNIN

Description of transient normal flow/unit width data read in for non-zero normal flow external boundaries from UNIT 20.

Variable	Type	Description
FTIMINC	Real	Time increment (secs) between consecutive sets of normal flow boundary condition values contained in this file.
NFLBN	Integer	The total number of flow boundary nodes that were designated in the UNIT 14 file as having a non zero normal flow.
QNIN(I)	Real	Normal flow/unit width (e.g., m^2/s) at specified normal flow node I. A positive flow/unit width is into the domain and a negative flow/unit width is out of the domain. The sequencing is assumed to match what is defined in the part of the UNIT 14 file specifying non-zero normal flow boundaries.

NOTES:
- This file is only used if one or more specified (non-zero) normal flow external boundaries were defined in the UNIT 14 input, (IBTYPE=2, 12 or 22) and NFFR = 0 in the UNIT 15 file.
- The first set of normal flow values is provided at TIME=STATIM. Subsequent sets of normal flow values are provided every FTIMINC
- Enough sets of normal flow/unit width values must be provided to extend for the entire model run, otherwise the run will crash!

5.2.5.6 *File: fort.21*

Input variables listed by the line sequence within the file:

AFRIC
JKI, FRIC(JKI) JKI=1,NP

Description of input variables read in from UNIT 21.

Variable	Type	Description
AFRIC	Real	Alphanumeric friction file ID (<= 24 characters).
JKI, FRIC(JKI) JKI=1,NP	Integer, Real	Node number, nodal bottom friction coefficient. Nodal values must be read in in ascending order.

5.2.5.7 File: fort.22

Input variables listed by the line sequence within the file:

```
IF(NWS.EQ.1) NHG,WSX2(NHG),WSY2(NHG), PR2(NHG), NHG=1,NP
IF(NWS.EQ.2) NHG,WSX2(NHG),WSY2(NHG), PR2(NHG), NHG=1,NP
IF(NWS.EQ.3)
  IWTIME
  WSPEED(I,J)
  WDIR(I,J)
IF(NWS.EQ.4)
  NHG, WSX2(NHG),WSY2(NHG), PR2(NHG)
```

Description of wind input data read in from UNIT 22.

Variable	Type	Description
NHG	Integer	Node number.
PR2(NHG)	Real	If NWS = 1 or 2, Applied atmospheric pressure at the free surface (N/m^2 = PA) divided by the reference density of water divided by gravity. If NWS = 4, Applied atmospheric pressure at the free surface (millibars).
WSX2(NHG)	Real	If NWS = 1, or 2, Applied horizontal free surface stress in the X-direction divided by the reference density of water at the node (should be units $(length/time)^2$). If NWS = 4, Applied horizontal wind velocity (knots) blowing toward the + X-direction.
WSY2(NHG)	Real	If NWS = 1or 2, Applied horizontal free surface stress in the Y-direction divided by the reference density of water at the node (should be units $(length/time)^2$). If NWS = 4, Applied horizontal wind velocity (knots) blowing toward the + Y-direction.
IWTIME		If NWS = 3, Time of the wind field in the following integer format: YEAR*1000000 + MONTH*10000 + DAY*100 + HR.
WSPEED(I,J)		If NWS = 3, Wind speed in meter/sec.
WDIR(I,J)		If NWS = 3, Direction wind blows from in degrees CW from north.

Notes:

NWS = 1

- The first data set is provided at TIME=STATIM+DT. Subsequent data sets are provided at every time step.

- Data must be provided for the entire model run, otherwise the run will crash!!!

NWS = 2

- The first data set is provided at TIME=STATIM. Subsequent data sets are provided every wind time interval.
- Data must be provided for the entire model run, otherwise the run will crash!!!
- The wind time interval must be set in the UNIT 15 file.

NWS = 3

- The first data set must be at or before the date and time listed in the UNIT 15 file as the beginning time of the simulation.
- Data sets are provided every WTIMINC, where this parameter is the wind time interval and is specified in the UNIT 15 file.
- Data must be provided for the entire model run, otherwise the run will crash!!!
- Values for NWLAT, NWLON, WTIMINC, the beginning time of the and several other parameters must be set in the unit 15 file.
- The following transformations are preformed to put this information into usable form for the model calculations.

  WIND_STRESS = DRAG_COEFF*0.001293*WIND_VEL*WIND_SPEED
  DRAG_COEFF = 0.001*(0.75+0.067*WIND_SPEED)
  IF(DRAG_COEFF.GT.0.003) DRAG_COEFF=0.003

NWS = 4

- This line must have the format I8,3E13.5.
- This line is repeated for as many nodes as desired.
- A line containing the # symbol in column 2 indicates new wind and pressure fields (i.e., values at the next time increment) begin on the following line.
- Each node that is not contained in the UNIT 22 file is assumed to have zero wind velocity and pressure = 1013.0.
- The following transformations are preformed to put this information into usable form for the model calculations.

  WIND_VEL (M/S) = WSX2*1.04*0.5144, WSY2*1.04*0.5144
  WIND_STRESS = DRAG_COEFF*0.001293*WIND_VEL*WIND_SPEED
  DRAG_COEFF = 0.001*(0.75+0.067*WIND_SPEED)
  IF(DRAG_COEFF.GT.0.003) DRAG_COEFF=0.003
  PR2*100/GRAVITY/1000.
- The first data set is provided at TIME=STATIM. Subsequent data sets are provided every wind time interval.
- Data must be provided for the entire model run, otherwise the run will crash!!!
- The wind time interval must be set in the UNIT 15 file.

5.2.5.8 File: fort. 23

Input variables listed by the line sequence within the file:

NODE #, AMP(I, NODE #), PHASE(I, NODE #)

Description of input variables read in from UNIT 23.

Variable	Type	Description
AMP(I, NODE)	Real	Amplitude of self-attraction/load tide constituent.
I	Integer	Index of tidal frequency.
NODE	Integer	Node number.
NP	Integer	Number of nodes in the grid.
PHASE(I, NODE)	Real	Phase in degrees of the self attraction/load tide constituent
FREQUENCY	Real	Frequency in rad/sec of the self attraction/load tide constituent.

Notes: This file is only used if NTIP=2. The format is identical to the "TEA" harmonic format with no velocity information included. Information is grouped by constituents that are assumed to be in the same order as the tidal potential terms listed in the UNIT 15 file. Phases must be in degrees. Amplitudes must be in a unit compatible with the units of gravity. These values are modified by the nodal factor and equilibrium argument provided for the tidal potential terms.

5.2.5.9 File: fort.200, fort.206, fort.212,...

Input variables listed by the line sequence within the file:

LENPDS, LENKGDS, NWORDS
GRBDATA

Description of input variables read in from UNIT 200, 206, 212, ...

Variable	Type	Description
GRBDATA(1,J)	Real	X-component of wind velocity at 10 m (cm/s).
GRBDATA(2,J)	Real	Y-component of wind velocity at 10 m (cm/s).
GRBDATA(3,J)	Real	Surface pressure (millibars).
NWORDS	Integer	Number of nodes in Gaussian grid.

Notes:

NWLAT Number of latitude values in MET file.
= 190 for type 126 global Gaussian grid.
NWLON Number of longitude values in MET file.
= 384 for type 126 global Gaussian grid.

WTIMINC Time interval (in sec) between MET files.
= 6.*3600 = 21600 for AVN files.

- The first data set must be 6 hours after the beginning of a cold start (UNIT 206) or at the beginning of a hot start (UNIT 200).
- Data must be provided for the entire model run, otherwise the run will crash!!!
- The following transformations are preformed to put this information into usable form for the model calculations:

wind_stress = drag_coeff*0.001293*wind_vel*wind_speed
drag_coeff = 0.001*(0.75+0.067*wind_speed)
if(drag_coeff.gt.0.003) drag_coeff=0.003
PR2/gravity/1000.

5.2.5.10 File: fort.200, fort.201, fort.202,...

Input variables listed by the line sequence within the file:

IYEAR, IMONTH, IDAY, IHOUR
UE(I, J), VE(I, J), PE(I, J)

Description of input variables read in from UNIT 200, 201, 202, ...

Variable	Type	Description
I,J	Integer	Longitude, latitude counters in E grid.
PE(I,J)	Real	Surface pressure (millibars).
UE(I,J)	Real	X-component of wind velocity at 10 m (cm/s in E grid orientation).
VE(I,J)	Real	Y-component of wind velocity at 10 m (cm/s in E grid orientation).

Notes:

NWLAT Number of latitude values in MET file.
= 271 for standard ETA-29 grid.
NWLON Number of longitude values in MET file.
= 181 for standard ETA-29 grid.
WTIMINC Time interval (in sec) between MET data.
= 3.*3600 = 10800 for ETA-29 files.

- The first data set must be 3 hours after the beginning of a cold start (UNIT 201) or at the beginning of a hot start (UNIT 200).
- Data must be provided for the entire model run, otherwise the run will crash!!!
- The following transformations are preformed to put this information into usable form for the model calculations:

Wind_stress = drag_coeff*0.001293*wind_vel*wind_speed

Drag_coeff = 0.001*(0.75+0.067*wind_speed)
If (drag_coeff.gt.0.003) drag_coeff=0.003
PR2/gravity/1000

5.2.6 Output File Variable Descriptions

5.2.6.1 File: fort.51

Output variables listed by the line sequence within the file:

NHARFR
HAFREQ, HAFF, HAFACE, HAFNAM
NSTAE
J, AMP(I,J), PHASE(I,J)

Description of output variables written to UNIT 51 (ASCII).

Variable	Type	Description
AMP(I, J)	Real	Elevation amplitude (in units of distance consistent with gravity).
HAFACE(K) K=1,NHARFR	Real	Phase equilibrium argument (in deg).
HAFF(K) K=1,NHARFR	Real	Amplitude nodal factor.
HAFNAM	Real	Constituent name.
HAFREQ(K), K=1,NHARFR	Real	Frequency (rad/s).
I	Integer	Frequency number.
J	Integer	Station number.
NHARFR	Integer	Number of frequencies harmonically analyzed.
NSTAE	Integer	Number of elevation stations included in the analysis.
PHASE(I, J)	Real	Elevation phase (in deg).

5.2.6.2 *File: fort.52*

Output variables listed by the line sequence within the file:

NHARFR
HAFREQ, HAFF, HAFACE, HAFNAM
NSTAV
J, UAMP(I,J), UPHASE(I,J),VAMP(I,J),VPHASE(I,J)

Description of output variables written to UNIT 52 (ASCII).

Variable	Type	Description
HAFACE(K) K=1,NHARFR	Real	Phase equilibrium argument (in deg).
HAFF(K) K=1,NHARFR	Real	Nodal factor.
HAFNAM	Real	Constituent name.
HAFREQ(K), K=1,NHARFR	Real	Frequency (rad/s).
I	Integer	Frequency number.
J	Integer	Node number.
NHARFR	Integer	Number of frequencies harmonically analyzed.
NSTAV	Integer	Number of velocity stations included in the analysis.
UAMP(I,J)	Real	X-direction velocity amplitude (in units of distance consistent with gravity).
UPHASE(I,J)	Real	X-direction velocity phase (in deg).
VAMP(I,J)	Real	Y-direction velocity amplitude (in units of distance consistent with gravity).
VPHASE(I,J)	Real	Y-direction velocity phase (in deg).

5.2.6.3 *File: fort.53*

Output variables listed by the line sequence within the file:

NHARFR
HAFREQ, HAFF, HAFACE, HAFNAM
NP
J, AMP(I,J), PHASE(I,J)

Description of output variables written to UNIT 53 (ASCII).

Variable	Type	Description
AMP(I,J)	Real	Elevation amplitude (in units of distance consistent with gravity).
HAFACE(K) K=1,NHARFR	Real	Phase equilibrium argument (in deg).
HAFF(K) K=1,NHARFR	Real	Amplitude nodal factor.
HAFNAM	Real	Constituent name.
HAFREQ(K), K=1,NHARFR	Real	Frequency (rad/s).
I	Integer	Frequency number.
J	Integer	Node number.
NHARFR	Integer	Number of frequencies harmonically analyzed.
NP	Integer	Number of nodes in grid.
PHASE(I,J)	Real	Elevation phase (in deg).

5.2.6.4 File: fort.54

Output variables listed by the line sequence within the file:

NHARFR
HAFREQ, HAFF, HAFACE, HAFNAM
NP
J, UAMP(I,J), UPHASE(I,J), VAMP(I,J), VPHASE(I,J)

Description of output variables written to UNIT 54 (ASCII).

Variable	Type	Description
HAFACE(K) K=1,NHARFR	Real	Phase equilibrium argument (in deg).
HAFF(K) K=1,NHARFR	Real	Amplitude nodal factor.
HAFNAM	Real	Constituent name.
HAFREQ(K), K=1,NHARFR	Real	Frequency (rad/s).
I	Integer	Frequency number.
J	Integer	Node number.
NHARFR	Integer	Number of frequencies harmonically analyzed.
NP	Integer	Number of nodes in grid

Variable	Type	Description
UAMP (I,J),	Real	X-direction velocity amplitude (in units of distance consistent with gravity).
UPHASE (I,J)	Real	X-direction velocity phase (in deg).
VAMP (I,J),	Real	Y-direction velocity amplitude (in units of distance consistent with gravity).
VPHASE (I,J)	Real	Y-direction velocity phase (in deg).

5.2.6.6 File: fort.55

Output variables listed by the line sequence within the file:

NP
J, TSEA(J), HAEA(J), HAEA(J)/TSEA(J), TSEV(J), HAEV(J), HAEV(J)/TSEV(J), J=1,NP
J, TSUA(J), HAUA(J), HAUA(J)/TSUA(J), TSUV(J), HAUV(J), HAUV(J)/TSUV(J)
TSVA(J), HAVA(J), HAVA(J)/TSVA(J) ,TSVV(J), HAVV(J), HAVV(J)/TSVV(J), J=1,NP

Description of output variables written to UNIT 55 (ASCII).

Variable	Type	Description
HAEA	Real	Mean elevation in the resynthesized time series.
HAEV	Real	Elevation variance in the resynthesized time series.
HAUA	Real	Mean X-velocity in the resynthesized time series.
HAUV	Real	X-velocity variance in the resynthesized time series.
HAVA	Real	Mean Y-velocity in the resynthesized time series.
HAVV	Real	Y-velocity variance in the resynthesized time series.
J	Integer	Node number.
NP	Integer	Number of nodes in grid.
TSEA	Real	Mean elevation in the raw model time series.
TSEV	Real	Elevation variance in the raw model time series.
TSUA	Real	Mean X-velocity in the raw model time series.
TSUV	Real	X-velocity variance in the raw model time series.
TSVA	Real	Mean Y-velocity in the raw model time series.
TSVV	Real	Y-velocity variance in the raw model time series.

5.2.6.7 File: fort.61

Output variables listed by the line sequence within the file:

RUNDES, RUNID, AGRID
NTRSPE, NSTAE, DT*NSPOOLE, NSPOOLE, IRTYPE
TIME, IT
I, ET00(I)

Note: The station number is not included if the file is binary.

Description of output variables written to UNIT 61.

Variable	Type	Description
DT*NSPOOLE	Real	Time increment between data sets (sec).
ET00(I), I = 1, NSTAE	Real	Surface elevation at NSTAE elevation recording stations.
I	Integer	Station number at NSTAE elevation recording stations.
IRTYPE	Integer	= 1 (the record type).
IT	Integer	Time step at which results at elevation stations are provided.
NSPOOLE	Integer	Time step increment between data sets.
NSTAE	Integer	Number of stations within each data set.
NTRSPE	Integer	Number of data sets to be spooled to UNIT 61.
RUNDES, RUNID, AGRID	Real	Alphanumeric descriptors that are described in UNIT 14 and UNIT 15 input descriptions.
TIME	Real	Time (in seconds) at which results at elevation stations are provided.

5.2.6.8 File: fort.62

Output variables listed by the line sequence within the file:

RUNDES, RUNID, AGRID
NTRSPV, NSTAV, DT*NSPOOLV, NSPOOLV, IRTYPE
TIME, IT
I, UU00(I), VV00(I)

Note: The station number is not included if the file is binary.

Description of output variables written to UNIT 62.

Variable	Type	Description
DT*NSPOOLV	Real	Time increment between data sets (sec).
I	Integer	Station number.
IRTYPE	Integer	= 2 (the record type).
IT	Integer	Time step at which results at velocity stations are provided.
NSPOOLV	Integer	Time step increment between data sets.
NSTAV	Integer	Number of stations within each data set.
NTRSPV	Integer	Number of data sets to be spooled to UNIT 62.
RUNDES, RUNID, AGRID	Real	Alphanumeric descriptors that are described in UNIT 14 and UNIT 15 input descriptions.
TIME	Real	Time (in seconds) at which results at velocity stations are provided.
UU00(I), I = 1, NSTAV	Real	X-velocity at NSTAV velocity recording stations.
VV00(I), I = 1, NSTAV	Real	Y-velocity at NSTAV velocity recording stations.

5.2.6.9 File: fort.63

Output variables listed by the line sequence within the file:

RUNDES, RUNID, AGRID
NDSETSE, NP, DT*NSPOOLGE, NSPOOLGE, IRTYPE
TIME, IT
I, ETA2(I)

Note: The station number is not included if the file is binary.

Description of output variables written to UNIT 63.

Variable	Type	Description
DT*NSPOOLGE	Real	The time increment between data sets (sec).
ETA2(I), I=1,NP	Real	Surface elevation for all nodes in the domain.
I	Integer	Node number (Note: The node number is not included if the file is binary).
IRTYPE	Integer	= 1 (the record type).
IT	Integer	Time step at which results at all nodes are provided.
NDSETSE	Integer	The number of data sets to be spooled to UNIT 63.

Variable	Type	Description
NP	Integer	The number of node points within each data set.
NSPOOLGE	Integer	The time step increment between data sets.
RUNDES, RUNID, AGRID	Real	Alphanumeric descriptors that are described in UNIT 14 and UNIT 15 input descriptions.
TIME	Real	Time (in seconds) at which results at all nodes are provided.

5.2.6.10 File: fort.64

Output variables listed by the line sequence within the file:

RUNDES, RUNID, AGRID
NDSETSV, NP, DT*NSPOOLGV, NSPOOLGV, IRTYPE
TIME, IT
I, UU2(I), VV2(I)

Note: The station number is not included if the file is binary.

Description of output variables written to UNIT 64.

Variable	Type	Description
DT*NSPOOLGV	Real	Time increment between data sets (sec).
I	Integer	Node number.
IRTYPE	Integer	= 2 (the record type).
IT	Integer	Time step at which results at all nodes are provided.
NDSETSV	Integer	Number of data sets to be spooled to UNIT 64.
NP	Integer	Number of node points within each data set.
NSPOOLGV	Integer	Time step increment between data sets.
RUNDES, RUNID, AGRID	Real	Alphanumeric descriptors that are described in UNIT 14 and UNIT 15 input descriptions.
TIME	Real	Time (in seconds) at which results at all nodes are provided.
UU2(I), I=1,NP	Real	X-velocity for all nodes in the domain.
VV2(I), I=1,NP	Real	Y-velocity for all nodes in the domain.

5.2.6.11 File: fort.71

Output variables listed by the line sequence within the file:

RUNDES,RUNID,AGRID
NTRSPC, NSTAC, DT*NSPOOLC, NSPOOLC, IRTYPE
TIME, IT
I, CC00(I)

Note: The station number is not included if the file is binary.

Description of output variables written to UNIT 71.

Variable	Type	Description
CC00(I), I=1,NSTAC	Real	Concentration at NSTAC concentration recording stations.
DT*NSPOOLC	Real	Time increment between data sets (sec).
I	Integer	Station number.
IRTYPE	Integer	= 1 (the record type).
IT	Integer	Time step at which results at concentration stations are provided.
NSPOOLC	Integer	Time step increment between data sets.
NSTAC	Integer	Number of stations within each data set.
NTRSPC	Integer	Number of data sets to be spooled to UNIT 71.
RUNDES, RUNID, AGRID	Real	Alphanumeric descriptors that are described in UNIT 14 and UNIT 15 input descriptions.
TIME	Real,	Time (in seconds) at which results at concentration stations are provided.

5.2.6.12 File: fort.73

Output variables listed by the line sequence within the file:

RUNDES, RUNID, AGRID
NDSETSC, NP, DT*NSPOOLGC, NSPOOLGC, IRTYPE
TIME, IT
I, C1(I)

Note: The node number is not included if the file is binary.

Description of output variables written to UNIT 73.

Variable	Type	Description
C1(I), I=1, NP	Real	Concentration for all nodes in the domain.
DT*NSPOOLGC	Real	The time increment between data sets (sec).
I	Integer	Node number.
IRTYPE	Integer	= 1 (the record type).
IT	Integer	Time step at which results at all nodes are provided.
NDSETSC	Integer	The number of data sets to be spooled to UNIT 73.
NP	Integer	The number of node points within each data set.
NSPOOLGC	Integer	The time step increment between data sets.
RUNDES, RUNID, AGRID	Real	Alphanumeric descriptors that are described in UNIT 14 and UNIT 15 input descriptions.
TIME	Real	Time (in seconds) at which results at all nodes are provided.

5.2.6.13 File: fort.74

Output variables listed by the line sequence within the file:

RUNDES, RUNID, AGRID
NDSETSW, NP, DT*NSPOOLGW, NSPOOLGW, IRTYPE
TIME, IT
I, WSX2(I), WSY2(I)

Note: The node number is not included if the file is binary.

Description of output variables written to UNIT 74.

Variable	Type	Description
DT*NSPOOLGW	Real	Time increment between data sets (sec).
I	Integer	Node number.
IRTYPE	Integer	= 2 (the record type).
IT	Integer	Time step at which results at all nodes are provided.
NDSETSW	Integer	Number of data sets to be spooled to UNIT 74.
NP	Integer	Number of node points within each data set.
NSPOOLGW	Integer	Time step increment between data sets.
RUNDES, RUNID, AGRID	Real	Alphanumeric descriptors that are described in UNIT 14 and UNIT 15 input descriptions.
TIME	Real	Time (in seconds) at which results at all nodes are provided.
WSX2(I), I=1, NP	Real	X wind stress for all nodes in the domain.
WSY2(I), I=1, NP	Real	Y wind stress for all nodes in the domain.

5.2.6.14 File: fort.67/68

Input variables listed by the line sequence within the file:

```
   IM
   TIME, IT
   ETA1(I)
   ETA2(I)
   UU1(I)
   VV1(I)
   IF(IM.EQ.10) CH1(I)
   NNODECODE
   IESTP, NSCOUE
   IVSTP, NSCOUV
   ICSTP, NSCOUC
   IGEP, NSCOUGE
   IGVP, NSCOUGV
   IGCP, NSCOUGC
   IGWP, NSCOUGW
IF (IHARIND.EQ.1)
   ICHA
   NZ, NF, MM, NP, NSTAE, NSTAV, NHASE, NHASV, NHAGE, NHAGV, ICALL
   NHARFR-NF
   HAFNAM(I),HAFREQ(I),HAFF(I),HAFACE(I), I=1,NHARFR
   TIMEUD
   A(I,J), I=1,MM, J=1,MM
   IF (NHASE.EQ.1) (STAELV(I,N) I=1,MM, N=1,NSTAE)
   IF (NHASV.EQ.1) (STAULV(I,N), STAVLV(I,N) I=1,MM, N=1,NSTAV)
   IF (NHAGE.EQ.1) (GLOELV(I,N) I=1,MM, N=1,NGLOE)
   IF (NHAGV.EQ.1) (GLOULV(I,N), GLOVLV(I,N) I=1,MM, N=1,NGLOV)
   IF (FMV.GT.0.)
      NTSTEPS
      ELAV(I), ELVA(I), I=1,NP
      XVELAV(I),YVELAV(I), XVELVA(I),YVELVA(I) I=1,NP
```

Description of output variables written to UNITS 67 and 68 (binary).

Variable	Type	Description
CH1(I)	Real	Depth-averaged scalar concentration value at node k at the current time step.
ELAV(I) I=1,NP	Real	Sum of elevations computed by ADCIRC, at every node I in the model grid, over all time steps since harmonic analysis means and variance checking has begun.
ELVA(I), I=1,NP	Real	Sum of squares of elevations computed by ADCIRC, at every node I in the model grid, over all time steps since harmonic analysis means and variance checking has begun.

Variable	Type	Description (fort.67/68, Continued)
ETA1(I)	Real	Surface elevation at node I at the previous time step.
ETA2(I)	Real	Surface elevation at node I at the current time step.
GLOELV(I,N) I=1,MM N=1,NGLOE	Real	Harmonic analysis global elevation load vector at all nodes in the model grid.
GLOULV(I,N) I=1,MM N=1,NGLOV	Real	Harmonic analysis global X-velocity load vector at all nodes in the model grid.
GLOVLV(I,N) I=1,MM N=1,NGLOV	Real	Harmonic analysis global Y-velocity load vector at all nodes in the model grid.
HAFACE(I) I=1,NHARFR	Real	Phase equilibrium argument (in deg).
HAFF(I) =1,NHARFR	Real	Amplitude nodal factor.
HAFNAM(I) =1,NHARFR	Real	Constituent name.
HAFREQ(I) =1,NHARFR	Real	Frequency (rad/s).
ICALL	Integer	Number of times the harmonic analysis has been updated.
ICHA	Integer	Time step counter to determine when the next update will be made to the harmonic analysis.
ICSTP, NSCOUC	Integer, Integer	Line number (for ASCII output) or record number (for binary output) of the most recent entry and time step counter to determine when the next entry will be written to the UNIT 81 output file.
IESTP, NSCOUE	Integer, Integer	Line number (for ASCII output) or record number (for binary output) of the most recent entry, and time step counter to determine when the next entry will be written in UNIT 61 output file.
IGCP, NSCOUGC	Integer, Integer	Line number (for ASCII output) or record number (for binary output) of the most recent entry and time step counter to determine when the next entry will be written to the UNIT 83 output file.
IGEP, NSCOUGE	Integer, Integer	Line number (for ASCII output) or record number (for binary output) of the most recent entry and time step counter to determine when the next entry will be written in the UNIT 63 output file.
IGVP, NSCOUGV	Integer, Integer	Line number (for ASCII output) or record number (for binary output) of the most recent entry and time step counter to determine when the next entry will be written to the UNIT 64

Variable	Type	Description (fort.67/68, Continued)
		output file.
IGWP, NSCOUGW	Integer, Integer	Line number (for ASCII output) or record number (for binary output) of the most recent entry and time step counter to determine when the next entry will be written to the UNIT 73 and UNIT 74 output files.
IM	Integer	Type of model run.
IT	Integer	Time step of hot start file.
IVSTP, NSCOUV	Integer, Integer	Line number (for ASCII output) or record number (for binary output) of the most recent entry and time step counter to determine when the next entry will be written in the UNIT 62 output file.
MM	Integer	= 2*NHARFR – NF.
NF	Integer	Indicator of whether the steady frequency is included in the harmonic analysis (NF = 1, steady is included; NF = 0, steady is not included).
NHAGE	Integer	Parameter that describes the spatial locations where harmonic analysis is performed (See fort.15 description, p.23).
NHAGV	Integer	Parameter that describes the spatial locations where harmonic analysis is performed (See fort.15 description, p.23).
NHARFR	Integer	Number of frequencies harmonically analyzed.
NHASE	Integer	Parameter that describes the spatial locations where harmonic analysis is performed (See fort.15 description, p.24).
NHASV	Integer	Parameter that describes the spatial locations where harmonic analysis is performed (See fort.15 description, p.24).
NNODECODE(I)	Integer	Node code at node I indicating whether the node is presently wet (active) or dry (inactive).
NP	Integer	Number of nodes in the grid.
NSTAE	Integer	Number of elevation recording stations.
NSTAV	Integer	Number of velocity recording stations.
NTSTEPS	Integer	Number of time steps since harmonic analysis means and variance checking has begun.
NZ	Integer	Indicator of whether the steady frequency is included in the harmonic analysis (NZ = 0, steady is included; NZ = 1, steady is not included).
STAELV(I,N) I=1,MM N=1,NSTAE	Real	Harmonic analysis station elevation load vectors at elevation recording stations.
STAULV(I,N) I=1,MM N=1,NSTAV	Real, Real	Harmonic analysis station X-velocity load vector at velocity recording stations.
STAVLV(I,N) I=1,MM N=1,NSTAV		Harmonic analysis station Y-velocity load vector at velocity recording stations.

Variable	Type	Description (fort.67/68, Continued)
TIME	Real	Time (in seconds) of hot start file.
TIMEUD	Real	Model time when the harmonic analysis was last updated.
UU1(I)	Real	Depth-averaged velocity in the X-coordinate direction at node I at the current time step.
VV1(I)	Real	Depth-averaged velocity in the Y-coordinate direction at node I at the current time step.
XVELAV(I), YVELAV(I) I=1,NP	Real, Real	Sum of depth-averaged U and V velocities computed by ADCIRC, at every node I in the model grid, over all time steps since harmonic analysis means and variance checking has begun.
XVELVA(I), YVELVA(I) I=1,NP	Real, Real	Sum of squares of depth-averaged U and V velocities computed by ADCIRC, at every node I in the model grid, over all time steps since harmonic analysis. Means and variance checking has begun.

5.2.7 Conversion Subroutines

5.2.7.1 Subroutine CPP

CPP transforms longitude and latitude (lamda, phi) coordinates into Cartesian coordinates using a Carte Parallelogrammatique Projection (CPP). Longitude and latitude must be in radians. CPP is called by the main code.

Calling Sequence: Subroutine CPP (X, Y, RLAMBDA, PHI, RLAMBDA0, PHI0)

Data Declaration: Real X, Y, RLAMBDA, PHI, RLAMBDA0, PHI0

Arguments:

X	X-coordinate in CPP projection.
Y	Y-coordinate in CPP projection.
RLAMBDA	Longitude (in radians).
PHI	Latitude (in radians).
RLAMBDA0	Reference longitude for CPP projection.
PHI0	Reference latitude for CPP projection.

5.2.7.2 Subroutine INVCP

This routine transforms Cartesian CPP coordinates to longitude and latitude (lambda, phi) coordinates. Longitude and latitude are given in radians. INVCP is called by the main code.

Calling Sequence: Subroutine INVCP (XXCP, YYCP, RLAMBDA, PHI, RLAMBDA0, PHI0)

Data Declaration: Real XXCP, YYCP, RLAMBDA, PHI, RLAMBDA0, PHI0

Arguments:

XXCP	X-coordinate in CPP projection.
YYCP	Y-coordinate in CPP projection.
RLAMBDA	Longitude (in radians).
PHI	Latitude (in radians).
RLAMBDA0	Reference longitude for CPP projection.
PHI0	Reference latitude for CPP projection.

5.2.7.3 Subroutine NEIGHB

NEIGHB generates a neighbor table from an element connectivity table. NEIGHB is called by the main code.

Calling Sequence: Subroutine NEIGHB (NE, NP, NNEIGH, NEIGH, NEIMIN, NEIMAX, NSCREEN)

Data Declaration: Integer NE, NP, NNEIGH, NEIGH, NEIMIN, NEIMAX, NSCREEN

Input Parameters:

MNP	Maximum number of nodal points.
MNE	Maximum number of elements.
MNEI	1 + maximum number of nodes connected to any one node in the finite element grid.

Arguments:

NE	Number of elements.
NP	Number of nodes in grid.
NNEIGH	Number of neighbors for each node.
NEIGH	2D array of neighbors for each node.
NEIMIN	1+ minimum number of neighbors for any node.
NEIMAX	1+ maximum number of neighbors for any node.
NSCREEN	Parameter which controls output to UNIT 6.

Note: The node itself is listed as neighbor #1. All other neighbors are sorted and placed in CW order from east.

5.2.7.4 Subroutine TIMECONV

TIMECONV converts time from year, month, day, hour, min, and sec into seconds since the beginning of the year specified in the arguments, IYR. TIMECONV is called by the main code.

Calling Sequence: Subroutine TIMECONV (IYR, IMO, IDAY, IHR, IMIN, SEC, TIMESEC)

Data Declaration: Real SEC, TIMESEC
Integer IYR, IMO, IDAY, IHR, IMIN

Arguments:

IYR	Year specified in Fleet Numeric Wind file.
IMO	Month specified in Fleet Numeric Wind file.
IDAY	Day specified in Fleet Numeric Wind file.
IHR	Hour specified in Fleet Numeric Wind file.
IMIN	Minute specified in Fleet Numeric Wind file.
SEC	Second specified in Fleet Numeric Wind file.
TIMESEC	Total seconds since beginning of the year, specified by IYR.

5.2.8 Wind and Pressure Field Subroutines

5.2.8.1 Subroutine NWS3GET

NWS3GET reads in wind fields from U.S. Navy Fleet Numeric product wind files and interpolates them onto the ADCIRC grid. Surface pressure is set to zero for this forcing option (Note that this format for Fleet Numeric data is obsolete.). NWS3GET is called by the main code.

Calling Sequence: Subroutine NWS3GET (X, Y, SLAM, SFEA, WVNX, WVNY, IWTIME, IWYR, WTIMED, NP, NWLON, NWLAT, WLATMAX, WLONMIN, WLATINC, WLONINC, ICS)

Data Declaration: Real X, Y, SLAM, SFEA, WVNX, WVNY, WTIMED, WLATMAX, WLONMIN, WLATINC, WLONINC
Integer ITWYR, NP, NWLON, NWLAT, ICS

Input Parameters:

MNWLAT	Maximum number of latitudes in Fleet Numeric Wind file.
MNWLON	Maximum number of longitudes in Fleet Numeric Wind file.

Arguments:

X	X-coordinate in the CPP projection.
Y	Y-coordinate in the CPP projection.
SLAM	Longitude on which CPP projection is centered (in degrees).
SFEA	Latitude on which CPP projection is centered (in degrees).
WVNX	Horizontal wind speed (m/s).
WVNY	Horizontal wind direction (deg.).
IWTIME	Time in U.S. Navy fleet wind file.
IWYR	Year in U.S. Navy fleet wind file.
WTIMED	Seconds since the beginning of the year.
NP	Number of nodal points in grid.
NWLON	Number of longitude values in wind file.
NWLAT	Number of latitude values in wind file.
WLATMAX	Maximum latitude (deg) of data in the UNIT 22 file (<0 south of the equator).
WLONMIN	Minimum longitude (deg) of data in the UNIT 22 file (<0 west of Grenich meridian).
WLATINC	Latitude increment (deg) of data in the UNIT 22 file (must be >0).
WLONINC	Longitude increment (deg) of data in the UNIT 22 file (must be >0).
ICS	Parameter that determines which coordinate system the calculations are performed.

Note: The ADCIRC grid information consists only of the longitudes and latitudes of the nodes. THE LONGITUDES AND LATITUDES MUST BE IN RADIANS! All wind speeds are converted to m/s and all pressures to m of water before they are returned.

5.2.8.2 Subroutine NWS4GET

NWS4GET reads wind fields from the PBL-JAG model onto the ADCIRC grid. NWS4GET is called by the main code.

Calling Sequence: Subroutine NWS4GET (WVNX, WVNY, PRN, NP, RHOWAT, G)

Data Declaration:
Real WVNX, WVNY, PRN, RHOWAT, G
Integer NP

Arguments:

WVNX	Surface wind speed values in the X-direction (m/s).
WVNY	Surface wind speed values in the Y-direction (deg).
PRN	Surface pressure at all nodes (Note: Units start in millibars of water and are converted to meters of water).
NP	Number of nodal points in grid.

RHOWAT	Reference density of water kg/m^3.
G	Gravitational constant.

Note: Output from this subroutine is X-component of wind velocity, Y-component of wind velocity (m/s) and pressure (m of water) on the ADCIRC grid. Background pressure is assumed to be 1013 mbars.

5.2.8.3 Subroutine NWS10GET

NWS10GET reads in wind fields form the U.S. National Weather Service AVN model SFLUX meteorological files and interpolates them onto the ADCIRC grid. NWS10GET is called by the main code.

Calling Sequence: Subroutine NWS10GET(NWSGGWI, FLON, FLAT, ULL, VLL, PLL, NP, RHOWAT, G, LONB, LATB)

Data Declaration: Real FLON, FLAT, ULL, VLL, PLL, RHOWAT, G
Integer NWSGGWI, NP, LONB, LATB

Input Parameters:

MNWLAT	LATB is defined as 190 for Gaussian grid.
MNWLON	LONB is defined as 384 for Gaussian grid.

Arguments:

NWSGGWI	Index number.
FLON	Longitude of node in the ADCIRC grid.
FLAT	Latitude of node in the ADCIRC grid.
ULL	Wind speed in X-direction (m/s).
VLL	Wind speed in Y-direction (m/s).
PLL	Surface pressure at all nodes (initially N/m^2 of water converted to meters of water).
NP	Number of nodal points in grid.
RHOWAT	Reference density of water kg/m^3.
G	Gravitational constant.
LONB	Number of longitude values in wind file (=NWLON).
LATB	Number of latitude values in wind file (=NWLAT).

Notes: The input files are in binary and have been created by the GRIB unpacking program unpkgrb1.f to extract only the X-component of wind velocity at 10 m, Y-component of wind velocity at 10 m, and surface pressure fields.
The SFLUX files utilize a global Gaussian Longitude/latitude grid that is constructed in these subroutines.
The ADCIRC grid information consists only of the longitude and latitude of the nodes. THE LONGITUDES AND LATITUDES MUST BE IN RADIANS!
Output from this subroutine is X-component of wind velocity at 10 m, Y-component of wind velocity at 10 m (m/s) and pressure (m of water) on the ADCIRC grid.

5.2.8.4 Subroutine GLATS

GLATS computes the latitudes in a Global Gaussian Longitude/latitude grid with T126 resolution (GRIB Grid type 126). GLATS is called by NWS10GET.

Calling Sequence: Subroutine GLATS (LGGHAF, COLRAD, WGT, WGTCS, RCS2)

Data Declaration: Real COLRAD, WGT, WGTCS, RCS2
Integer LGGHAF

Arguments:

LGGHAF	Half the latitude dimension of the Gaussian grid.
COLRAD	
WGT	
WGTCS	
RCS2	

5.2.8.5 Subroutine POLY

The subroutine is used by GLATS.

Calling Sequence: Subroutine POLY(N, RAD, P)

Data Declaration: Real RAD, P
Integer N

Arguments: N
RAD
P

5.2.8.6 Subroutine G2RINI

G2RINI computes the factors to interpolate from a global Gaussian Longitude/latitude grid with T126 resolution (GRIB Grid type 126) onto another grid. The new grid is a series of longitude and latitude points contained in the FLON and FLAT arrays with a total number of points NP. G2RINI is called by NWS10GET.

Calling Sequence: Subroutine G2RINI (GCLON, GCLAT, FLON, FLAT, N00, N10, N11 N01, D00, D10, D11, D01, NP, LONB, LATB)

Data Declaration: Real GCLON, GCLAT, FLON, FLAT, D00, D10, D11, D01
Integer N00,N10,N11,N01,NP

Arguments:	GCLON	
	GCLAT	
	FLON	Longitude of node in the ADCIRC grid.
	FLAT	Latitude of node in the ADCIRC grid.
	N00	
	N10	
	N11	
	N01	
	D00	
	D10	
	D11	
	D01	
	NP	Number of nodal points in grid.
	LONB	Number of longitude values in wind file (=NWLON).
	LATB	Number of latitude values in wind file (=NWLAT).

5.2.8.7 Subroutine NWS11GET

NSW11GET reads in wind fields from U.S. National Weather Service ETA-29 model that have been stripped down and given to us by NOAA and interpolates the data onto the ADCIRC grid The input files are in binary and have been created by NOAA and contain only the X-component of wind velocity at 10 m, Y-component of wind velocity at 10m (m/s) and surface pressure fields (mbars). The ETA-29 model uses an E grid and therefore the X and Y-components of wind velocity are not oriented along lines of constant latitude and longitude. These must be converted to be useful in ADCIRC. NWS11GET is called by the main code.

Calling Sequence:	Subroutine NWS11GET (NWSEGWI, IDSETFLG, FLON, FLAT, ULL, VLL, PLL, NP, RHOWAT, G)	
Input Parameters:	MNWLAT	LATB defined as 271 for ETA-29 grid.
	MNWLON	LONB defined as 181 for ETA-29 grid.
Data Declaration:	Real FLON, FLAT, ULL, VLL, PLL, RHOWAT, G Integer NWSEGWI, NP	
Arguments:	NWSEGWI	Index number.
	IDSETFLG	Flag to compute only E29 interpolating factors (=0).
	FLON	Longitude of node in the ADCIRC grid.
	FLAT	Latitude of node in the ADCIRC grid.
	ULL	Interpolated wind vector in X-direction at all nodes (m/s).
	VLL	Interpolated wind vector in Y-direction at all

	nodes (m/s).
PLL	Surface pressure at all nodes (initially millibars, converted to meters of water).
NP	Number of nodal points in grid.
RHOWAT	Reference density of water kg/m^3.
G	Gravitational constant.

Notes: Output from this subroutine is X-component of wind velocity, Y-component of wind velocity (m/s) and surface pressure (m water) on the ADCIRC grid.

5.2.8.8 Subroutine E29SEARCH

E29SEARCH is a routine that finds where a given longitude, latitude falls in the ETA 29 grid and determines the interpolating factors to interpolate ETA 29 fields to that position. It then computes the angle to rotate the ETA 29 velocity field in order to convert to a longitude, lat coordinate system. This routine is called by NWS11GET.

Calling Sequence: Subroutine E29SEARCH (NODE, FLON, FLAT, NN1, NN2, NN3, DD1, DD2, DD3, BETAU)

Data Declaration:
Real FLON,FLAT,DD1,DD2,DD3,BET
Integer NODE, NN1, NN2, NN3

Arguments:

NODE	ADCIRC grid node number.
FLON	Longitude of node in the ADCIRC grid.
FLAT	Latitude of node in the ADCIRC grid.
NN1, NN2, NN3, DD1, DD2, DD3	Interpolating factors for ETA29 grid.
BETAU	Conversion angle between E29 velocity field and a longitude/latitude coordinate system.

5.2.8.9 Subroutine E29CALC

This subroutine computes the longitude and latitude of a given I, J position in the ETA 29 grid. E29CALC is called by the subroutines E29SEARCH and BETAUCALC.

Calling Sequence: Subroutine E29CALC (I, J, LAMDA, PHI, N)

Data Declaration:
Real PHI
Integer I, J, LAMDA, N

Arguments:		
	I	Longitude indice in ETA29 grid.
	J	Latitude indice in ETA29 grid.
	LAMDA	Longitude value at point I.
	PHI	Latitude value at point J.
	N	Dimension of point I, J.

5.2.8.10 Subroutine BETAUCALC

BETAUCALC computes the conversion angle between the ETA29 velocity field and a longitude, lat coordinate system. BETAUCALC is called by E29SEARCH.

Calling Sequence: Subroutine BETAUCALC (I1, J1, DD1, I2, J2, DD2, I3, J3, DD3, BETAU)

Data Declaration: Real DD1, DD2, DD3, BETAU
Integer I1, J1, I2, J2, I3, J3

Arguments:		
	I1, J2	Indices of triangle, corner 1.
	DD1	length of triangle side, 1.
	I2, J2	Indices of triangle, corner 2.
	DD2	length of triangle side, 2.
	I3, J3	Indices of triangle, corner 3.
	DD3	length of triangle side, 3.
	BETAU	Conversion angle between E29 velocity field and a longitude/latitude coordinate system.

5.2.9 Matrix Solver Subroutines

5.2.9.1 Subroutine LSQUPDLHS

This subroutine updates the left-hand side matrix. LSQUPDLHS is called by the main code.

Calling Sequence: Subroutine LSQUPDLHS(TIME, IT)

Data Declaration: Real TIME, IT

Arguments:		
	TIME	Absolute model time (sec).
	IT	Model time step.

Common Blocks: LSQFREQS
LSQPARMS
MATRIX

5.2.9.2 Subroutine LSQUPDES

This subroutine updates the right-hand side load vectors for the elevation station harmonic analysis. LSQUPDES is called by the main code.

Calling Sequence: Subroutine LSQUPDES (STAE, NSTAE)

Data Declaration: Real STAE
Integer NSTAE

Arguments:

STAE	Station elevation values used to update load vectors.
NSTAE	Number of tidal elevation recording stations.

Common Blocks: LSQFREQS
LSQPARMS
LOADVECES

5.2.9.3 Subroutine LSQUPDVS

LSQUPDVS updates the right-hand side load vectors for the velocity station harmonic analysis. LSQUPDVS is called by the main code.

Calling Sequence: Subroutine LSQUPDVS (STAU, STAV, NSTAV)

Data Declaration: Real STAU, STAV, NSTAV

Arguments:

STAU	Station X-component velocity values used to update load vectors.
STAV	Station Y-component velocity values used to update load vectors.
NSTAV	Number of tidal current recording stations.

Common Blocks: LSQFREQS
LSQPARMS
LOADVECVS

5.2.9.4 Subroutine LSQUPDEG

Subroutine LSQUPDEG updates the right-hand side load vectors for the global elevation harmonic analysis. LSQUPDEG is called by the main code.

Calling Sequence: Subroutine LSQUPDEG (GLOE, NP)

Data Declaration:
Real GLOE
Integer NP

Arguments:

GLOE	Global elevation values used to update load vectors.
NP	Number of points in global grid.

Common Blocks:
LSQFREQS
LSQPARMS
LOADVECEG

5.2.9.5 Subroutine LSQUPDVG

LSQUPDVG updates the right-hand side load vectors for the global velocity harmonic analysis. LSQUPDVG is called by the main code.

Calling Sequence: Subroutine LSQUPDVG (GLOU, GLOV, NP)

Data Declaration:
Real GLOU, GLOV
Integer NP

Arguments:

GLOU	Global U velocity values used to update load vectors.
GLOV	Global V velocity values used to update load vectors.
NP	Number of points in global grid.

Common Blocks:
LSQFREQS
LSQPARMS
LOADVECVG

5.2.9.6 Subroutine FULSOL

Subroutine FULSOL fills out, decomposes and solves the LSQ system. Solves system A*X=B by L*D*L(TR) decomposition in full storage mode. FULSOL is called by the main code, LSQSOLES, LSQSOLVS, LSQSOLEG, and LSQSOLVG.

Calling Sequence: Subroutine FULSOL (IDECOM)

Data Declaration: Integer IDECOM

Arguments: IDECOM Flag to fill out matrix and decompose only (=0).

Common Blocks: LSQPARMS
MATRIX

5.2.9.7 Subroutine LSQSOLES

LSQSOLES solves the system and writes output for elevation stations. LSQSOLES is called by the main code.

Calling Sequence: Subroutine LSQSOLES (NSTAE)

Data Declaration: Integer NSTAE

Arguments: NSTAE Number of elevation recording stations.

Common Blocks: LSQFREQS
LSQPARMS
LOADVECES
MATRIX

Notes: NF = 0 if no steady constituent.
NF = 1 if steady constituent.

5.2.9.8 Subroutine LSQSOLVS

Subroutine LSQSOLVS solves the system and writes output for velocity stations. LSQSOLVS is called by the main code.

Calling Sequence: Subroutine LSQSOLVS (NSTAV)

Data Declaration: Integer NSTAV

Arguments: NSTAV Number of velocity recording stations.

Common Blocks: LSQFREQS
LSQPARMS
LOADVECVS
MATRIX

Notes: NF = 0 if no steady constituent.
NF = 1 if steady constituent.

5.2.9.9 Subroutine LSQSOLEG

LSQSOLEG solves the system and writes output for elevation globally. LSQSOLEG is called by the main code.

Calling Sequence: Subroutine LSQSOLEG (NP)

Data Declaration: Integer NP

Arguments: NP Number of nodal points in grid.

Common Blocks: LSFREQS
LSQPARMS
LOADVECEG
MATRIX

Notes: NF = 0 if no steady constituents.
NF = 1 if steady constituents.

5.2.9.10 Subroutine LSQSOLVG

LSQSOLVG solves the system and writes output for velocity globally. LSQSOLVG is called by the main code.

Calling Sequence: Subroutine LSQSOLVG (NP)

Data Declaration: Integer NP

Arguments: NP Number of nodal points in grid.

Common Blocks: LSQFREQS
LSQPARMS
LOADVECVG
MATRIX

Notes: NF = 0 if no steady constituents.
NF = 1 if steady constituents.

5.2.10 Harmonic Analysis Subroutines

5.2.10.1 Subroutine HAHOUT

HAHOUT writes out to the hot start file (UNITS 67 and 68) header information and the left hand side matrix for the harmonic analysis. HAHOUT is called by the main code.

Calling Sequence: Subroutine HAHOUT (NP, NSTAE, NSTAV, ISTAE, ISTAV, IGLOE, IGLOV, IOUNIT, IHOTSTP)

Data Declaration: Integer NP, NSTAE, NSTAV, ISTAE, ISTAV, IGLOE, IGLOV, IOUNIT,

IHOTSTP

Arguments:

NP	Number of nodal points in grid.
NSTAE	Number of elevation recording stations.
NSTAV	Number of velocity recording stations.
ISTAE	Harmonic analysis performance at elevation recording stations (=1).
ISTAV	Harmonic analysis performance at velocity recording stations (=1).
IGLOE	Harmonic analysis performance for global elevations (=1).
IGLOV	Harmonic analysis performance for global velocities (=1).
IOUNIT	UNIT number of hot start file (UNITS 67 or 68).
IHOTSTP	Starting record number in the hot start file.

Common Blocks: LSQFREQS
LSQPARMS
MATRIX

5.2.10.2 Subroutine HAHOUTES

HAHOUTES writes elevation station harmonic analysis right hand side load vector to a hot start file (UNITS 67 and 68). HAHOUTES is called by the main code.

Calling Sequence: Subroutine HAHOUTES (HSTAE, IOUNIT, IHOTSTP)

Data Declaration: Real HSTAE
Integer IOUNIT, IHOTSTP

Arguments:

HSTAE	Number of elevation recording stations.
IOUNIT	UNIT number of hot start file (UNITS 67 or 68).
IHOTSTP	Starting record number in the hot start file.

Common Blocks: LSQPARMS
LOADVECES

5.2.10.3 Subroutine HAHOUTVS

HAHOUTVS writes velocity station harmonic analysis right hand side load vector to a hot start file (UNITS 67 and 68). HAHOUTVS is called by the main code.

Calling Sequence: Subroutine HAHOUTVS (NSTAV, IOUNIT, IHOTSTP)

Data Declaration: Integer NSTAV, IOUNIT, IHOTSTP

Arguments:

NSTAV	Number of velocity recording stations.
IOUNIT	UNIT number of hot start file (67 or 68).
IHOTSTP	Starting record number in the hot start file.

Common Blocks: LSQPARMS
LOADVECVS

5.2.10.4 Subroutine HAHOUTEG

Subroutine HAHOUTEG writes global elevation harmonic analysis right hand side load vector to a hot start file (UNITS 67 and 68). HAHOUTEG is called by the main code.

Calling Sequence: Subroutine HAHOUTEG (NP, IOUNIT, IHOTSTP)

Data Declaration: Integer NP, IOUNIT, IHOTSTP

Arguments:

NP	Number of nodal points in grid.
IOUNIT	UNIT number of hot start file (UNITS 67 or 68).
IHOTSTP	Starting record number in the hot start file.

Common Blocks: LSQPARMS
LOADVECEG

5.2.10.5 Subroutine HAHOUTVG

HAHOUTVG writes global velocity harmonic analysis right hand side load vector to a hot start file (UNITS 67 and 68). HAHOUTVG is called by the main code.

Calling Sequence: Subroutine HAHOUTVG (NP, IOUNIT, IHOTSTP)

Data Declaration: Integer NP, IOUNIT, IHOTSTP

Arguments:

NP	Number of nodal points in grid.
IOUNIT	UNIT number of hot start file (UNITS 67 or 68).
IHOTSTP	Starting record number in the hot start file.

Common Blocks: LSQPARMS
LOADVECVG

5.2.10.6 Subroutine HACOLDS

HACOLDS initializes parameters for harmonic analysis with a cold start. HACOLDS is called by the main code.

Calling Sequence: Subroutine HACOLDS

Common Blocks: LSQFREQS
LSQPARMS
MATRIX

5.2.10.7 Subroutine HACOLDSES

HACOLDSES initializes the elevation station load vectors for harmonic analysis with a cold start. HACOLDSES is called by the main code.

Calling Sequence: Subroutine HACOLDSES (NSTAE)

Data Declaration: Integer NSTAE

Arguments:

NSTAE	Number of elevation recording stations.

Common Blocks: LSQPARMS
LOADVECES

5.2.10.8 Subroutine HACOLDSVS

HACOLDSVS initializes elevation station load vectors for harmonic analysis with a cold start. HACOLDSVS is called by the main code.

Calling Sequence: Subroutine HACOLDSVS (NSTAV)

Data Declaration: Integer NSTAV

Arguments: NSTAV Number of velocity recording stations.

Common Blocks: LSQPARMS
LOADVECVS

5.2.10.9 Subroutine HACOLDSEG

HACOLDSEG initializes global elevation load vectors for harmonic analysis with a cold start. HACOLDSEG is called by the main code.

Calling Sequence: Subroutine HACOLDSEG (NP)

Data Declaration: Integer NP

Arguments: NP Number of nodal points in grid.

Common Blocks: LSQPARMS
LOADVECEG

5.2.10.10 Subroutine HACOLDSVG

HACOLDSVG initializes global velocity load vectors for harmonic analysis with a cold start. HACOLDSVG is called by the main code.

Calling Sequence: Subroutine HACOLDSVG (NP)

Data Declaration: Integer NP

Arguments: NP Number of nodal points in grid.

Common Blocks: LSQPARMS
LOADVECVG

5.2.10.11 Subroutine HAHOTS

HAHOTS reads in and initializes harmonic analysis for a hot start. Checks are made within the subroutine to ensure there is agreement between values read in from the hot start file and values read in from the UNIT 15 file. HAHOTS is called by the main code.

Calling Sequence: Subroutine HAHOTS (INSTAE, INSTAV, INP, IISTAE, IISTAV, IIGLOE, IIGLOV, NSCREEN, IHOTSTP, IHOT)

Data Declaration: Integer INSTAE, INSTAV, INP, IISTAE, IISTAV, IIGLOE, IIGLOV, NSCREEN, IHOTSTP, IHOT)

Arguments:

INSTAE	Number of elevation recording stations.
INSTAV	Number of velocity recording stations.
INP	Number of nodal points in grid.
IISTAE	Harmonic analysis performance at elevation recording stations (=1).
IISTAV	Harmonic analysis performance at velocity recording stations (=1).
IIGLOE	Harmonic analysis performance for global elevations (=1).
IIGLOV	Harmonic analysis performance for global velocities (=1).
NSCREEN	Parameter which controls output to UNIT 6.
IHOTSTP	Starting record number in the hot start file.
IHOT	Parameter that controls whether the model is hot started.

Common Blocks: LSQFREQS
LSQPARMS
MATRIX

5.2.10.12 Subroutine HAHOTSES

HAHOTSES reads in and initializes the elevation station load vector for harmonic analysis with a hot start. HAHOTSES is called by the main code.

Calling Sequence: Subroutine HAHOTSES (NSTAE, IHOTSTP, IHOT)

Data Declaration: Integer NP, IOUNIT, IHOTSTP

Arguments:

NSTAE	Number of elevation recording stations.
IHOTSTP	Starting record number in the hot start file.
IHOT	Parameter that controls whether the model is hot started.

Common Blocks: LSQPARMS
LOADVECES

5.2.10.13 Subroutine HAHOTSVS

HAHOTSVS reads in and initializes the velocity station load vector for harmonic analysis with a hot start. HAHOTSVS is called by the main code.

Calling Sequence: Subroutine HAHOTSVS (NSTAV, IHOTSTP, IHOT)

Data Declaration: Integer NSTAV, IHOTSTP, IHOT

Arguments:

NSTAV	Number of velocity recording stations.
IHOTSTP	Starting record number in the hot start file.
IHOT	Parameter that controls whether the model is hot started.

Common Blocks: LSQPARMS
LOADVECVS

5.2.10.14 Subroutine HAHOTSEG

HAHOTSEG reads in and initializes the global elevation load vector for harmonic analysis with a hot start. HAHOTSEG is called by the main code.

Calling Sequence: Subroutine HAHOTSEG (NP, IHOTSTP, IHOT)

Data Declaration: Integer NP, IHOTSTP, IHOT

Arguments:

NP	Number of nodal points in grid.
IHOTSTP	Starting record number in the hot start file.
IHOT	Parameter that controls whether the model is hot started.

Common Blocks: LSQPARMS
LOADVECEG

5.2.10.15 Subroutine HAHOTSVG

HAHOTSVG reads in and initializes the global velocity load vector for harmonic analysis with a hot start. HAHOTSVG is called by the main code.

Calling Sequence: Subroutine HAHOTSVG (NP, IHOTSTP, IHOT)

Data Declaration: Integer NP, IHOTSTP, IHOT

Arguments:

NP	Number of nodal points in grid.
IHOTSTP	Starting record number in the hot start file.
IHOT	Parameter that controls whether the model is hot started.

Common Blocks: LSQPARMS
LOADVECVG

5.3 CSC Solver

5.3.1 Constraints and Limitations

The iterative solver, ITPACKV 2D, requires machine dependent modifications before it can be compiled and run. The most important are SRELPR, which is machine relative precision, and RPARM, which is stopping criterion. Also change system-dependent routine second. These modifications are located in routines DFAULT and TIMER. If a direct banded matrix solver from LINPACK is used to solve the GWCE, more memory is necessary to run the solver compared to the iterative solver. For example, on small problems, e.g. 1000 nodes, the iterative solver requires about ½ the memory of the direct solver. For larger problems, e.g. > 20,000 nodes, the iterative solver requires about 1/5 the memory of the direct solver.

5.3.2. ITPACKV 2D Iterative Solvers

5.3.2.1 Subroutine JCG

JCG drives the Jacobi conjugate gradient algorithm. JCG is called by the main code.

Calling Sequence: Subroutine JCG (N, NDIM, MAXNZ, JCOEF, COEF, RHS, U, IWKSP, NW, IPARM, WKSP, RPARM, IER)

Data Declaration: Integer N, NDIM, JCOEF, IWKSP, NW, IPARM, IER
Real MAXNZ, COEF, RHS, U, WKSP, RPARM

Arguments:	N	Input integer of dimension of the matrix.
	NDIM	Row dimension of JCOEF and COEF arrays in calling routine.
	MAXNZ	Maximum number of non-zeros per row.
	JCOEF	Integer array for sparse matrix representation.
	COEF	Array for sparse matrix representation. JCOEF and COEF use the ELLPACK data structure.
	RHS	Input vector that contains the right-hand side of the matrix problem.
	U	Input/output vector. On input, U contains initial guess to the solution. On output, it contains the latest estimate to the solution.
	IWKSP	Integer vector workspace of length 3*N.
	NW	Input integer. Length of available WKSP. On output, IPARM(8) is amount used.
	IPARM	Integer vector of length 12 that allows user to specify some integer parameters which affect the method.
	WKSP	Vector used for working space. Jacobi conjugate gradient needs this to be in length at least 4*N + 4*ITMAX. Here, ITMAX = IPARM(1) is the maximum allowable number of iterations.
	RPARM	Vector of length 12 that allows user to specify some parameters which affect the method.
	IER	Output integer. Error flag.

Common Blocks: ITCOM1
ITCOM2
ITCOM3

5.3.2.2 Subroutine JSI

Subroutine JSI drives the Jacobi semi-iteration algorithm. JSI is called by the main code.

Calling Sequence: Subroutine JSI (N, NDIM, MAXNZ, JCOEF, COEF, RHS, U, IWKSP, NW, IPARM, WKSP, RPARM, IER)

Data Declaration: Integer N, NDIM, JCOEF, IWKSP, NW, IPARM, IER
Real MAXNZ, COEF, RHS, U, WKSP, RPARM

Arguments:	N	Input integer of dimension of the matrix.
	NDIM	Row dimension of JCOEF and COEF arrays in calling routine.
	MAXNZ	Maximum number of non-zeros per row.
	JCOEF	Integer array for sparse matrix representation.

COEF	Array for sparse matrix representation. JCOEF and COEF use the ELLPACK data structure.
RHS	Input vector that contains the right-hand side of the matrix problem.
U	Input/output vector. On input, U contains initial guess to the solution. On output, it contains the latest estimate to the solution.
IWKSP	Integer vector workspace of length 3*N.
NW	Input integer. Length of available WKSP. On output, IPARM(8) is amount used.
IPARM	Integer vector of length 12 that allows user to specify some integer parameters which affect the method.
WKSP	Vector used for working space. Jacobi SI needs this to be in length at least 2*N.
RPARM	Vector of length 12 that allows user to specify some parameters which affect the method.
IER	Output integer. Error flag.

Common Blocks: ITCOM1
ITCOM2
ITCOM3

5.3.2.3 Subroutine SOR

Subroutine SOR drives the Successive over relaxation algorithm. SOR is called by the main code.

Calling Sequence: Subroutine SOR (N, NDIM, MAXNZ, JCOEF, COEF, RHS, U, IWKSP, NW, IPARM, WKSP, RPARM, IER)

Data Declaration: Integer N, NDIM, JCOEF, IWKSP, NW, IPARM, IER
Real MAXNZ, COEF, RHS, U, WKSP, RPARM

Arguments:

N	Input integer of dimension of the matrix.
NDIM	Row dimension of JCOEF and COEF arrays in calling routine.
MAXNZ	Maximum number of non-zeros per row.
JCOEF	Integer array for sparse matrix representation.
COEF	Array for sparse matrix representation. JCOEF and COEF use the ELLPACK data structure.
RHS	Input vector that contains the right-hand side of the matrix problem.
U	Input/output vector. On input, U contains initial guess to the solution. On output, it contains the latest estimate to the solution.
IWKSP	Integer vector workspace of length 3*N.

NW	Input integer. Length of available WKSP. On output, IPARM(8) is amount used.
IPARM	Integer vector of length 12 that allows user to specify some integer parameters which affect the method.
WKSP	Vector used for working space. SOR needs this to be in length at least 2*N.
RPARM	Vector of length 12 that allows user to specify some parameters which affect the method.
IER	Output integer. Error flag.

Common Blocks: ITCOM1
ITCOM2
ITCOM3

5.3.2.4 Subroutine SSORCG

SSORCG drives the Symmetric SOR-CG algorithm. SSORCG is called by the main code.

Calling Sequence: Subroutine SSORCG (N, NDIM, MAXNZ, JCOEF, COEF, RHS, U, IWKSP, NW, IPARM, WKSP, RPARM, IER)

Data Declaration: Integer N, NDIM, JCOEF, IWKSP, NW, IPARM, IER
Real MAXNZ, COEF, RHS, U, WKSP, RPARM

Arguments:

N	Input integer of dimension of the matrix.
NDIM	Row dimension of JCOEF and COEF arrays in calling routine.
MAXNZ	Maximum number of non-zeros per row.
JCOEF	Integer array for sparse matrix representation.
COEF	Array for sparse matrix representation. JCOEF and COEF use the ELLPACK data structure.
RHS	Input vector that contains the right-hand side of the matrix problem.
U	Input/output vector. On input, U contains initial guess to the solution. On output, it contains the latest estimate to the solution.
IWKSP	Integer vector workspace of length 3*N.
NW	Input integer. Length of available WKSP. On output, IPARM(8) is amount used.
IPARM	Integer vector of length 12 that allows user to specify some integer parameters which affect the method.
WKSP	Vector used for working space. SSORCG needs this to be in length at least 6*N + 4*ITMAX.
RPARM	Vector of length 12 that allows user to specify some parameters which affect the method.
IER	Output integer. Error flag.

Common Blocks: ITCOM1
ITCOM2
ITCOM3

5.3.2.5 Subroutine SSORSI

The subroutine SSORSI (Symmetric Successive Over- Relaxation Semi-Iteration) drives the symmetric SOR-SI algorithm. SSORSI is called by the main code.

Calling Sequence: Subroutine SSORSI (N, NDIM, MAXNZ, JCOEF, COEF, RHS, U, IWKSP, NW, IPARM, WKSP, RPARM, IER)

Data Declaration: Integer N, NDIM, JCOEF, IWKSP, NW, IPARM, IER
Real MAXNZ, COEF, RHS, U, WKSP, RPARM

Arguments:

N	Input integer of dimension of the matrix.
NDIM	Row dimension of JCOEF and COEF arrays in calling routine.
MAXNZ	Maximum number of non-zeros per row.
JCOEF	Integer array for sparse matrix representation.
COEF	Array for sparse matrix representation. JCOEF and COEF use the ELLPACK data structure.
RHS	Input vector that contains the right-hand side of the matrix problem.
U	Input/output vector. On input, U contains initial guess to the solution. On output, it contains the latest estimate to the solution.
IWKSP	Integer vector workspace of length 3*N.
NW	Input integer. Length of available WKSP. On output, IPARM(8) is amount used.
IPARM	Integer vector of length 12 that allows user to specify some integer parameters which affect the method.
WKSP	Vector used for working space. SSORSI needs this to be in length at least 5*N.
RPARM	Vector of length 12 that allows user to specify some parameters which affect the method.
IER	Output integer. Error flag.

Common Blocks: ITCOM1
ITCOM2
ITCOM3

5.3.2.6 Subroutine RSCG

The subroutine RSCG (Reduced System Conjugate Gradient) drives the Reduced System CG algorithm. RSCG is called by the main code.

Calling Sequence: Subroutine RSCG (N, NDIM, MAXNZ, JCOEF, COEF, RHS, U, IWKSP, NW, IPARM, WKSP, RPARM, IER)

Data Declaration: Integer N, NDIM, JCOEF, IWKSP, NW, IPARM, IER
Real MAXNZ, COEF, RHS, U, WKSP, RPARM

Arguments:

N	Input integer of dimension of the matrix.
NDIM	Row dimension of JCOEF and COEF arrays in calling routine.
MAXNZ	Maximum number of non-zeros per row.
JCOEF	Integer array for sparse matrix representation.
COEF	Array for sparse matrix representation. JCOEF and COEF use the ELLPACK data structure.
RHS	Input vector that contains the right-hand side of the matrix problem.
U	Input/output vector. On input, U contains initial guess to the solution. On output, it contains the latest estimate to the solution.
IWKSP	Integer vector workspace of length 3*N.
NW	Input integer. Length of available WKSP. On output, IPARM(8) is amount used.
IPARM	Integer vector of length 12 that allows user to specify some integer parameters which affect the method.
WKSP	Vector used for working space. RSCG needs this to be in length at least N+3*NB+4*ITMAX. Here, ITMAX = IPARM(1) and NB is the order of the black subsystem. ITMAX is the maximum allowable number of iterations.
RPARM	Vector of length 12 that allows user to specify some parameters which affect the method.
IER	Output integer. Error flag.

Common Blocks: ITCOM1
ITCOM2
ITCOM3

5.3.2.7 Subroutine RSSI

RSSI (Reduced System Semi-Iterative) drives the Reduced System SI algorithm. RSSI is called by the main code.

Calling Sequence: Subroutine RSSI (N, NDIM, MAXNZ, JCOEF, COEF, RHS, U, IWKSP, NW, IPARM, WKSP, RPARM, IER)

Data Declaration: Integer N, NDIM, JCOEF, IWKSP, NW, IPARM, IER
Real MAXNZ, COEF, RHS, U, WKSP, RPARM

Arguments:

N	Input integer of dimension of the matrix.
NDIM	Row dimension of JCOEF and COEF arrays in calling routine.
MAXNZ	Maximum number of non-zeros per row.
JCOEF	Integer array for sparse matrix representation.
COEF	Array for sparse matrix representation. JCOEF and COEF use the ELLPACK data structure.
RHS	Input vector that contains the right-hand side of the matrix problem.
U	Input/output vector. On input, U contains initial guess to the solution. On output, it contains the latest estimate to the solution.
IWKSP	Integer vector workspace of length 3*N.
NW	Input integer. Length of available WKSP. On output, IPARM(8) is amount used.
IPARM	Integer vector of length 12 that allows user to specify some integer parameters which affect the method.
WKSP	Vector used for working space. RSSI needs this to be in length at least N + NB. NB is the order of the black subsystem.
RPARM	Vector of length 12 that allows user to specify some parameters which affect the method.
IER	Output integer. Error flag.

Common Blocks: ITCOM1
ITCOM2
ITCOM3

5.3.3 ITPACKV 2D Solver Subroutines

5.3.3.1 Subroutine ITJCG

ITJCG performs one iteration of the Jacobi Conjugate Gradient algorithm. ITJCG is called by the subroutine JCG.

Calling Sequence: Subroutine ITJCG (N, NDIM, MAXNZ, JCOEF, COEF, U, U1, D, D1, DTWD, TRI)

Data Declaration: Integer N, NDIM, MAXNZ, JCOEF
Real COEF, U, U1, D, D1, DTWD, TRI

Arguments:

N	Input integer. Dimension of the matrix.
NDIM	Row dimension of JCOEF and COEF arrays in calling routine.
MAXNZ	Maximum number of non-zeros per row.
JCOEF	Integer sparse matrix representation.
COEF	Sparse matrix representation.
U	Input vector that contains the value of the solution vector at the end of in iterations.
U1	Input/output vector. On input, it contains the value of the solution at the end of the in-1 iteration. On output, it will contain the estimate for the solution vector.
D	Input vector that contains the pseudo-residual vector after in iterations.
D1	Input/output vector. On input, D1 contains the pseudo-residual vector after in-1 iterations. On output, it will contain the newest pseudo-residual vector.
DTWD	Array that is used in the computations of the acceleration parameter gamma and the new pseudo-residual.
TRI	Array that stores the tri-diagonal matrix associated with the Eigen values of the conjugate gradient polynomial.

Common Blocks: ITCOM1
ITCOM2
ITCOM3

5.3.3.2 Subroutine ITJSI

The subroutine ITJSI performs one iteration of the Jacobi Semi-Iterative algorithm. It is called by JSI.

Calling Sequence: Subroutine ITJSI (N, NDIM, MAXNZ, JCOEF, COEF, RHS, U, U1, D, ICNT)

Data Declaration:
Integer N, NDIM, MAXNZ JCOEF, ICNT
Real MAXNZ, COEF, RHS, U, U1, D

Arguments:

N	Input integer. Dimension of the matrix.
NDIM	Row dimension of JCOEF and COEF arrays in calling routine.
MAXNZ	Maximum number of non-zeros per row.
JCOEF	Integer sparse matrix representation.
COEF	Sparse matrix representation.
RHS	Input vector that contains the right-hand side of the matrix problem.
U	Input vector. Contains the value of the solution vector at the end of in iterations.
U1	Input/output vector. On input, it contains the value of the solution at the end of the in-1 iteration. On output, it will contain the estimate for the solution vector.
D	Input vector that contains the pseudo-residual vector after in iterations.
ICNT	Number of iterations since last change of SME.

Common Blocks: ITCOM1
ITCOM2
ITCOM3

5.3.3.3 Subroutine ITSOR

ITSOR performs one iteration of the Successive Over-relaxation algorithm. It is called by SOR.

Calling Sequence: Subroutine ITSOR (N, NB, NDIM, MAXNZ, JCOEF, COEF, RHS, U, NUP, MOVED, WK, IWK)

Data Declaration:
Integer N, NB, NDIM, MAXNZ, JCOEF, NUP, MOVED, IWK
Real COEF, RHS, U, WK

Arguments:	N	Input integer. Dimension of the matrix.
	NB	Dimension of the black subsystem.
	NDIM	Row dimension of JCOEF and COEF arrays in calling routine.
	MAXNZ	Maximum number of non-zeros per row.
	JCOEF	Integer sparse matrix representation.
	COEF	Sparse matrix representation.
	RHS	Input vector that contains the right-hand side of the matrix problem.
	U	Input vector that contains the value of the solution vector at the end of in iterations.
	NUP	Maximum number of non-zeros per row in upper triangle u (for natural ordering only).
	MOVED	Logical variable indicating whether or not was reshuffled.
	WK	Array. Work vector of length 2*N.
	IWK	Integer array giving wave populations.

Common Blocks: ITCOM1
ITCOM2
ITCOM3

5.3.3.4 Subroutine ITSRCG

Subroutine ITSRCG performs one iteration of the Symmetric SOR Conjugate Gradient algorithm. It is called by SSORCG.

Calling Sequence: Subroutine ITSRCG (N, NB, NUP, NLOW, NDIM, MAXNZ, JCOEF, COEF, RHS, U, U1, C, C1, D, DL, MOVED, WK, TRI, IWK)

Data Declaration: Integer N, NB, NUP, NLOW, NDIM, JCOEF, IWKSP, NW, MOVED, IWK
Real MAXNZ, COEF, RHS, U, U1, C, C1, D, DL, WK, TRI

Arguments:	N	Input integer. Dimension of the matrix.
	NB	Dimension of the black subsystem.
	NUP	Maximum number of non-zeros per row in upper triangle u (for natural ordering only).
	NLOW	Maximum number of non-zeros per row in the lower triangle l (input for natural ordering only).
	NDIM	Row dimension of JCOEF and COEF arrays in calling routine.
	MAXNZ	Maximum number of non-zeros per row.
	JCOEF	Integer sparse matrix representation.
	COEF	Sparse matrix representation.

	RHS	Input vector that contains the right-hand side of the matrix problem.
	U	Input vector that contains the value of the solution vector at the end of in iterations.
	U1	Input/output vector. On input, it contains the value of the solution at the end of the in-1 iteration. On output, it will contain the estimate for the solution vector.
	C	Input vector that contains the forward residual after in iterations.
	C1	Input/output vector. On input, C1 contains the forward residual after in-1 iterations. On output, C1 contains the updated forward residual.
	D	Input vector that contains the pseudo-residual vector after in iterations.
	DL	Vector that is used in the computations of the acceleration parameters.
	MOVED	Logical variable indicating whether or not was reshuffled.
	WK	Array. Work vector of length 2*N.
	TRI	Vector that stores the tri-diagonal matrix associated with the conjugate gradient acceleration.
	IWK	Integer array giving wave populations.

Common Blocks: ITCOM1
ITCOM2
ITCOM3

5.3.3.5 Subroutine ITSRSI

ITSRSI performs one iteration of the symmetric SOR Semi-iteration algorithm. It is called by SSORSI.

Calling Sequence: ITSRSI (N, NB, NUP, NLOW, NDIM, MAXNZ, JCOEF, COEF, RHS, U, U1, C, D, CTWD, MOVED, WK, IWK)

Data Declaration: Integer N, NB, NUP, NLOW, NDIM, MAXNZ, JCOEF, MOVED
Real COEF, RHS, U, U1, C, D, CTWD, WK, IWK

Arguments:		
	N	Input integer. Dimension of the matrix.
	NB	Dimension of the black subsystem.
	NUP	Maximum number of non-zeros per row in upper triangle u (for natural ordering only).
	NLOW	Maximum number of non-zeros per row in the lower triangle l (input for natural ordering only).
	NDIM	Row dimension of JCOEF and COEF arrays in calling routine.

MAXNZ	Maximum number of non-zeros per row.
JCOEF	Integer sparse matrix representation.
COEF	Sparse matrix representation.
RHS	Input vector that contains the right-hand side of the matrix problem.
U	Input vector that contains the value of the solution vector at the end of in iterations.
U1	Input/output vector. On input, it contains the value of the solution at the end of the in-1 iteration. On output, it will contain the estimate for the solution vector.
C	Input vector that contains the forward residual after in iterations.
D	Input vector that contains the pseudo-residual vector after in iterations.
CTWD	Vector that is used in the computations of the acceleration parameters.
WK	Array. Work vector of length 2*N.
IWK	Integer array giving wave populations.

Common Blocks: ITCOM1
ITCOM2
ITCOM3

5.3.3.6 Subroutine ITRSCG

ITRSCG performs one iteration of the Reduced System Conjugate Gradient algorithm. It is called by RSCG.

Calling Sequence: Subroutine ITRSCG (N, NB, NDIM, MAXNZ, JCOEF, COEF, UB, UB1, DB, DB1, WB, TRI)

Data Declaration: Integer N, NB, NDIM, MAXNZ, JCOEF
Real COEF, UB, UB1, DB, DB1, WB, TRI

Arguments:

N	Input integer. Dimension of the matrix.
NB	Dimension of the black subsystem.
NDIM	Row dimension of JCOEF and COEF arrays in calling routine.
MAXNZ	Maximum number of non-zeros per row.
JCOEF	Integer sparse matrix representation.
COEF	Sparse matrix representation.
UB	Input vector that contains the estimate for the solution on the black points after in iterations.
UB1	Input/output vector. On input, UB1 contains the solution vector after in-1 iterations. On output, it will

	contain the newest estimate for the solution vector. This is only for the black points.
DB	Input array. DB(NRP1) contains the value of the current pseudo-residual on the black points.
DB1	Input/output array. DB1(NRP1) contains the pseudo-residual on the black points for the in-1 iteration on input. On output, it is for the IN+1 iteration.
WB	Array that is used for computations involving black vectors.
TRI	Array that stores the tri-diagonal matrix with conjugate gradient acceleration.

Common Blocks: ITCOM1
ITCOM2
ITCOM3

5.3.3.7 Subroutine ITRSSI

ITRSSI performs one iteration of the Reduced System Semi-iteration algorithm. It is called by RSSI.

Calling Sequence: Subroutine ITRSSI (N, NB, NDIM, MAXNZ, JCOEF, COEF, RHS, UB, UB1, DB)

Data Declaration: Integer N, NB, NDIM, MAXNZ, JCOEF
Real COEF, RHS, UB, UB1, DB

Arguments:

N	Input integer. Dimension of the matrix.
NB	Dimension of the black subsystem.
NDIM	Row dimension of JCOEF and COEF arrays in calling routine.
MAXNZ	Maximum number of non-zeros per row.
JCOEF	Integer sparse matrix representation.
COEF	Sparse matrix representation.
RHS	Input vector that contains the right-hand side of the matrix problem.
UB	Input vector that contains the estimate for the solution on the black points after in iterations.
UB1	Input/output vector. On input, UB1 contains the solution vector after in-1 iterations. On output, it will contain the newest estimate for the solution vector. This is only for the black points.
DB	Input array. DB(NRP1) contains the value of the current pseudo-residual on the black points.

Common Blocks: ITCOM1
ITCOM2
ITCOM3

5.3.3.8 Subroutine CHGCON

CHGCON computes the new estimate for the largest eigenvalue for Conjugate Gradient acceleration. CHGCON is called by the subroutines ITJCG, ITSRCG, and ITSRSI.

Calling Sequence: Subroutine CHGCON (LDT, TRI, GAMOLD, RHOOLD, IBMTH)

Data Declaration: Integer LDT, IBMTH
Real TRI, GAMOLD, RHOOLD

Arguments:

LDT	Leading dimension of TRI.
TRI	Tri-diagonal matrix associated with the eigen values of the conjugate gradient polynomial.
GAMOLD	Previous value of acceleration parameter.
RHOOLD	Previous value of acceleration parameter.
IBMTH	Indicator of basic method being accelerated by CG.

Common Blocks: ITCOM1
ITCOM2
ITCOM3

Notes:

IBMTH	= 1, Jacobi
	= 2, Reduced system
	= 3, SSOR

5.3.3.9 Subroutine CHGSI

CHGSI computes new Chebyshev acceleration parameters adaptively. CHGSI is called by the subroutines ITRSSI, ITJSI, and ITSRSI.

Calling Sequence: Subroutine CHGSI (DTNRM,IBMTH)

Data Declaration: Integer IBMTH
Real DTNRM

Arguments:

DTNRM	Numerator of Rayleigh Quotient
IBMTH	Indicator of basic method being accelerated by CG.

Common Blocks: ITCOM1
ITCOM2
ITCOM3

Notes: IBMTH = 1, Jacobi
= 2, Reduced system
= 3, SSOR

5.3.3.10 Subroutine DFAULT

DFAULT sets the default values of IPARM and RPARM. DFAULT is called by the main code.

Calling Sequence: Subroutine DFAULT (IPARM, RPARM)

Data Declaration:
Integer IPARM
Real RPARM

Arguments:

IPARM	Integer vector of length 12 that allows user to specify some integer parameters which affect the method.
RPARM	Vector of length 12 that allows user to specify some parameters which affect the method.

Common Blocks:
ITCOM1
ITCOM2
ITCOM3

5.3.3.11 Subroutine ECHALL

ECHALL initializes the ITPACKV common blocks from the information contained in IPARM and RPARM. ECHALL also prints the values of all the parameters in IPARM and RPARM. ECHALL is called by the iterative solver subroutines JCG, JSI, SOR, SSORCG, SSORSI, RSCG, and RSSI.

Calling Sequence: Subroutine ECHALL (N, NDIM, MAXNZ, JCOEF, COEF, RHS, IPARM, RPARM, ICALL)

Data Declaration:
Integer N, NDIM, JCOEF, IPARM, ICALL
Real MAXNZ, COEF, RHS, RPARM

Arguments:

N	Input integer. Dimension of the matrix.
NDIM	Row dimension of JCOEF and COEF arrays in calling routine.
MAXNZ	Maximum number of non-zeros per row.
JCOEF	Integer sparse matrix representation.
COEF	Sparse matrix representation.

RHS	Input vector that contains the right-hand side of the matrix problem.
IPARM	Integer vector of length 12 that allows user to specify some integer parameters which affect the method.
RPARM	Vector of length 12 that allows user to specify some parameters which affect the method.
ICALL	Indicator of which parameters are being printed.

Common Blocks: ITCOM1
ITCOM2
ITCOM3

Notes: ICALL = 1, initial parameters
ICALL = 2, final parameters

5.3.3.12 Subroutine EQRT1S

EQRT1S computes the smallest or largest M eigenvalues of a symmetric tri-diagonal matrix. As written, the routine computes the M smallest eigenvalues. To compute the M largest eigenvalues, reverse the sign of each element of D before and after calling the routine. In this case, ISW must equal zero. EQRT1S is called by the subroutine ECHALL.

Calling Sequence: Subroutine EQRT1S (D, E2, N, M, ISW IER)

Data Declaration: Integer N, M. ISW, IER
Real D, E2

Arguments:

D	Input vector of length N containing the diagonal elements of the matrix. The computed eigenvalues replace the first M components of the vector D in non- decreasing sequence, while the remaining components are lost.
E2	Input vector of length N containing the squares of the off-diagonal elements of the matrix. Input E2 is destroyed.
N	Input scalar containing the order of the matrix.
M	Input scalar containing the number of smallest eigenvalues desired (M is less than or equal to N).
ISW	Input scalar meaning as follows: ISW = 1 means that the matrix is known to be positive definite. ISW = 0 means that the matrix is not known to be positive definite.
IER	Error parameter. (output) Warning error: IER = 601 indicates that successive iterates to the K^{th} eigenvalue were not monotone increasing. The value K is stored in E2(1).

Terminal error:
IER = 602 indicates that ISW = 1 but matrix is not positive definite.

Common Blocks: ITCOM1
ITCOM2
ITCOM3

5.3.3.13 Subroutine ITERM

ITERM produces the iteration summary line at the end of each iteration. If level .GE. 4, the latest approximation to the solution will be printed. ITERM is called by the subroutines ITJCG, ITJSI, ITSOR, ITSRCG, ITSRSI, ITRSCG, and ITRSSI.

Calling Sequence: Subroutine ITERM (N,COEF,U,WK,IMTHD)

Data Declaration: Integer N, IMTHD
Real COEF, U, WK

Arguments:

N	Order of system or, for reduced system routines, order of black subsystem.
IMTHD	Indicator of method: = 1, JCG = 2, JSI = 3, SOR = 4, SSORCG = 5, SSORSI = 6, RSCG = 7, RSSI
COEF	Iteration matrix.
U	Solution estimate.
WK	Work array of length N.

Common Blocks: ITCOM1
ITCOM2
ITCOM3

5.3.3.14 Subroutine MOVE

MOVE sets up the I-U-L data structure for the SOR and SSOR methods when natural ordering is used. MOVE is called by the subroutines SOR, SSORCG, and SSORSI.

Calling Sequence: Subroutine MOVE (N, NDIM, MAXNZ, NLOW, NUP, JCOEF, COEF, LEVEL, NOUT, MOVED, IWKSP)

Data Declaration: Integer N, NDIM, NLOW, NUP, JCOEF, LEVEL, NOUT, MOVED, IWKSP
Real MAXNZ, COEF, RHS, U, WKSP, RPARM

Arguments:

N	Order of matrix.
NUP	Maximum number of non-zeros per row in upper triangle U (output).
NLOW	Maximum number of non-zeros per row in the lower triangle l (input for natural ordering only).
NDIM	Row dimension of JCOEF and COEF arrays in calling routine.
MAXNZ	Maximum number of non-zeros per row.
JCOEF	Integer data structure for coefficient columns.
COEF	Data structure for coefficients.
LEVEL	Level of output.
NOUT	Output device number.
MOVED	Logical constant indicating whether or not data structure has been reshuffled.
IWKSP	Integer workspace.

5.3.3.15 Subroutine OMEG

OMEG computes new values for CME, OMEGA, and SPECR for fully adaptive SSOR methods. OMEG is called by the subroutines SSORCG, SSORSI, ITSRCG, and ITSRSI.

Calling Sequence: Subroutine OMEG (DNRM,IFLAG)

Data Declaration: Integer IFLAG
Real DNRM

Arguments:

DNRM	Numerator of Rayleigh Quotient
IFLAG	Indicator of appropriate entry point

Common Blocks: ITCOM1
ITCOM2
ITCOM3

5.3.3.16 Subroutine PARCON

PARCON computes acceleration parameters for conjugate gradient acceleration methods. PARCON is called by the subroutines ITJCG, ITSRCG, and ITRSCG.

Calling Sequence: Subroutine PARCON (DTNRM, C1, C2, C3, C4, GAMOLD, RHOTMP, IBMTH)

Data Declaration: Integer IBMTH
Real DTNRM, C1, C2, C3, C4, GAMOLD, RHOTMP, IBMTH

Arguments:

DTNRM	Inner product of residuals.
C1	Output -- rho*gamma.
C2	Output – rho.
C3	Output -- 1-rho.
C4	Output -- rho*(1-gamma).
GAMOLD	Output -- value of gamma at preceding iteration.
RHOTMP	Last estimate for value of rho
IBMTH	Indicator of basic method being accelerated by CG = 1, Jacobi. = 2, Reduced system. = 3, SSOR.

Common Blocks: ITCOM1
ITCOM2
ITCOM3

5.3.3.17 Subroutine PARSI

PARSI computes acceleration parameters for semi-iterative accelerated methods. PARSI is called by the subroutines ITJSI, ITSRSI, and ITRSSI.

Calling Sequence: Subroutine PARSI (C1, C2, C3, IBMTH)

Data Declaration: Integer IBMTH
Real C1, C2, C3

Arguments:

C1, C2, C3	Output acceleration parameters.
IBMTH	Indicator of basic method being accelerated by SI. = 1, Jacobi. = 2, Reduced system. = 3, SSOR.

Common Blocks: ITCOM1
ITCOM2
ITCOM3

5.3.3.18 Subroutine PBSOR

PBSOR computes a backward SOR sweep on U. PBSOR is called by the subroutines ITSRCG and ITSRSI.

Calling Sequence: Subroutine PBSOR (N, NB, NDIM, MAXNZ, JCOEF, COEF, U, RHS, NUP, NLOW, MOVED, IPOP)

Data Declaration: Integer N, NB, NDIM, JCOEF, NUP, NLOW, MOVED, IPOP
Real MAXNZ, COEF, RHS, U

Arguments:

N	Order of system.
NB	Order of black subsystem.
NDIM	Row dimension of COEF and JCOEF arrays in calling routine.
MAXNZ	Maximum number of non-zeros per row.
JCOEF	Integer array of coefficient columns.
COEF	Array of coefficients.
U	Latest estimate of solution.
RHS	Right hand side of matrix problem.
NUP	Maximum number of non-zeros per row in lower triangle l (input for natural order only).
NLOW	Maximum number of non-zeros per row in upper triangle u (input for natural order only).
MOVED	Logical variable indicating whether or not matrix has been reshuffled.
IPOP	Integer vector giving the wave populations.

Common Blocks: ITCOM1
ITCOM3

5.3.3.19 Subroutine PERMAT

PERMAT takes the sparse matrix representation of the matrix stored in the arrays JCOEF and COEF and permutes both rows and columns, overwriting the previous structure. This routine is to be called after subroutine SCAL by subroutines JCG, SOR, SSORCG, SSORSI, RSCG, RSSI, and SORWAV.

Calling Sequence: Subroutine PERMAT (N, NDIM, MAXNZ, JCOEF, COEF, P, WORK, IWORK)

Data Declaration: Integer N, NDIM, JCOEF, IWORK
Real MAXNZ, COEF, P, WORK

Arguments:

N	Order of system.
NDIM	Row dimension of arrays JCOEF and COEF in the calling routine.
MAXNZ	Maximum number of non-zero entries per row.
JCOEF	Integer array for data.
COEF	Array for data structure coefficients.
P	Permutation vector.
WORK	Workspace of length N.
IWORK	Integer workspace of length N.

5.3.3.20 Subroutine PERROR

PERROR computes the residual, R = RHS - A*U. The user also has the option of printing the residual and/or the unknown vector depending on IDGTS. PERROR is called by the subroutines JCG, JSI, SOR, SSORCG, SSORSI, RSCG, and RSSI.

Calling Sequence: Subroutine PERROR (N, NDIM, MAXNZ, JCOEF, COEF, RHS, U, WORK, DIGIT1, DIGIT2, IDGTS)

Data Declaration:
Integer N, NDIM, JCOEF, IWORK, IDGTS
Real MAXNZ, COEF, RHS, U, WORK, DIGIT1, DIGIT2,

Arguments:

N	Dimension of matrix.
NDIM	Row dimension of JCOEF and COEF in calling routine.
MAXNZ	Maximum number of non-zeros per row.
JCOEF	Integer array of sparse matrix representation.
COEF	Array of sparse matrix representation.
RHS	Right hand side of matrix problem.
U	Latest estimate of solution.
WORK	Workspace vector of length 2*n.
DIGIT1	Output - Measure of accuracy of stopping test.
DIGIT2	Output - Measure of accuracy of solution.
IDGTS	Parameter controlling level of output.

If IDGTS < 1 or > 4, then no output.
= 1, then number of digits is printed, provided level .GE. 1.
= 2, then solution vector is printed, provided level .GE. 1.
= 3, then residual vector is printed, provided level .GE. 1.
= 4, then both vectors are printed, provided level .GE. 1.

Common Blocks: ITCOM1
ITCOM2
ITCOM3

5.3.3.21 Subroutine PERVEC

PERVEC permutes a vector as dictated by the permutation vector P. If P(I) = J, then V(J) gets V(I). PERVEC is called by the subroutine JCG.

Calling Sequence: Subroutine PERVEC (N, P, V, WORK)

Data Declaration: Integer N, P
Real V, WORK

Arguments:

N	Length of vectors P, V, and WORK.
P	Integer permutation vector.
V	Vector to be permuted.
WORK	Workspace vector of length N.

5.3.3.22 Subroutine PFSOR

PFSOR computes a forward SOR sweep on U. PFSOR is called by the subroutines SSORCG, ITSRCG, and ITSRSI.

Calling Sequence: Subroutine PFSOR (N, NB, NDIM, MAXNZ, JCOEF, COEF, U, RHS, NUP, MOVED, IPOP)

Data Declaration: Integer N, NB, NDIM, JCOEF, NUP, MOVED, IPOP
Real MAXNZ, COEF, U, RHS

Arguments:

N	Order of system.
NB	Order of black subsystem.
NDIM	Row dimension of COEF and JCOEF arrays in calling routine.
MAXNZ	Maximum number of non-zeros per row.
JCOEF	Integer array of coefficient columns.
COEF	Array of coefficients.
U	Latest estimate of solution.
RHS	Right hand side of matrix problem.
NUP	Maximum number of non-zeros per row in lower triangle L (input for natural order only).
MOVED	Logical variable indicating whether or not matrix has been reshuffled.
IPOP	Integer vector giving the wave populations.

Common Blocks: ITCOM1
ITCOM3

5.3.3.23 Subroutine PFSOR1

PFSOR1 computes a forward SOR sweep on U and computes the norm of the pseudo-residual vector. PFSOR1 is called by the subroutine ITSOR.

Calling Sequence: Subroutine PFSOR1 (N, NB, NDIM, MAXNZ, JCOEF, COEF, U, RHS, NUP, MOVED, WORK, IPOP)

Data Declaration: Integer N, NB, NDIM, MAXNZ, JCOEF, NUP, MOVED, IPOP
Real, COEF, U, RHS, WORK

Arguments:

N	Order of system.
NB	Order of black subsystem.
NDIM	Row dimension of COEF and JCOEF arrays in calling routine.
MAXNZ	Maximum number of non-zeros per row.
JCOEF	Integer array of coefficient columns.
COEF	Array of coefficients.
U	Latest estimate of solution.
RHS	Right hand side of matrix problem.
NUP	Maximum number of non-zeros per row in lower triangle L (input for natural order only).
MOVED	Logical variable indicating whether or not matrix has been reshuffled.
WORK	Workspace vector of length N.
IPOP	Integer vector giving the wave populations.

Common Blocks: ITCOM1
ITCOM3

5.3.3.24 Subroutine PJAC

PJAC performs one Jacobi iteration. PJAC is called by the subroutines JCG, ITJCG, ITJSI, ITSRCG, and ITSRSI.

Calling Sequence: Subroutine PJAC (N, NDIM, MAXNZ, JCOEF, COEF, U, RHS)

Data Declaration: Integer N, NDIM, MAXNZ, JCOEF
Real, COEF, U, RHS

Arguments:	N	Order of system.
	NDIM	Row dimension of arrays JCOEF and COEF in the calling routine.
	MAXNZ	Maximum number of nonzero entries per row.
	JCOEF	Integer array for data.
	COEF	Array for data structure coefficients.
	U	Estimate of solution of a matrix problem.
	RHS	On input -- contains the right hand side of the matrix problem. On output -- contains B*U + RHS where B = I – A and A has been scaled to have a unit diagonal.

5.3.3.25 Subroutine PMULT

PMULT computes C = A*B, a matrix-vector product. Matrix A is assumed to be stored in the COEF, JCOEF ELLPACK data structure and all entries in the column array JCOEF are assumed to be between 1 and N, inclusive. A is assumed to have a unit diagonal. PMULT is called by the subroutine PERROR.

Calling Sequence: Subroutine PMULT (N, NDIM, MAXNZ, JCOEF, COEF, B, C)

Data Declaration: Integer N, NDIM, MAXNZ, JCOEF
Real, COEF, B, C

Arguments:	N	Order of system.
	NDIM	Row dimension of arrays JCOEF and COEF in the calling routine.
	MAXNZ	Maximum number of non-zero entries per row.
	JCOEF	Integer array for data.
	COEF	Array for data structure coefficients.
	B	Multiplying vector of length N.
	C	Product vector of length N.

5.3.3.26 Subroutine PRBNDX

PRBNDX computes the red-black permutation vectors P (and its inverse IP) if possible. The algorithm is to mark the first node as red (arbitrary). All of its adjacent nodes are marked black and placed in a stack. The remainder of the code pulls the first node off the top of the stack and tries to type its adjacent nodes. The typing of the adjacent point is a five way case statement which is well commented (see do loop 100).

The array P is used both to keep track of the color of a node (red node is positive, black is negative) but also the father node that caused the color marking of that point. Since complete

information on the adjacency structure is hard to come by, this forms a link to enable the color change of a partial tree when a recoverable color conflict occurs.

The array IP is used as a stack to point to the set of nodes left to be typed that are known to be adjacent to the current father node.

This routine is to be called after the subroutine SCAL by the subroutines JCG, JSI, SOR, SSORCG, RSCG, and RSSI.

Calling Sequence: Subroutine PRBNDX (N, NDIM, MAXNZ, JCOEF, P, IP, NBLACK, LEVEL, NOUT, IER)

Data Declaration: Integer N, NDIM, MAXNZ, JCOEF, IP, NBLACK, LEVEL, NOUT, IER
Real P

Arguments:

N	Number of nodes. (integer, scalar).
NDIM	Row dimension of JCOEF in calling routine.
MAXNZ	Maximum number of non-zero entries per row.
JCOEF	Array of column indices. It is assumed that for every row where only one element is stored, that element corresponds to the diagonal entry. The diagonal must be the first entry stored. (integer, arrays).
P, IP	Permutation and inverse permutation vectors. (integer, arrays each of length N).
NBLACK	Number of black nodes. Number of red nodes is N - NBLACK. (integer, scalar).
LEVEL	Switch for printing.
NOUT	Output tape number.
IER	Error flag. (integer, scalar) = 0, Normal return. Indexing performed successfully. = 201, Red-black indexing not possible.

5.3.3.27 Subroutine PRSBLK

PRSBLK computes a BLACK-RS sweep on a red vector into a black vector. PRSBLK is called by the subroutines RSCG, ITRSCG, and ITRSSI.

Calling Sequence: Subroutine PRSBLK (NB, NR, NDIM, MAXNZ, JCOEF, COEF, U, V)

Data Declaration: Integer NB, NR, NDIM, MAXNZ, JCOEF
Real COEF, U, V

Arguments:	NB	Number of black points.
	NR	Number of red points.
	NDIM	Row dimension of JCOEF and COEF arrays in calling routine.
	MAXNZ	Maximum number of non-zeros per row.
	JCOEF	Integer data structure for coefficient columns.
	COEF	Data structure for array coefficients.
	U	Latest estimate of the solution.
	V	On input -- Contains the right hand side. On output -- V(NR+1,...N) contains (FR**T)*UR+CB.

5.3.3.28 Subroutine PRSRED

PRSRED computes a RED-RS sweep on a black vector into a red vector. PRSRED is called by the subroutines RSCG, RSSI, ITRSCG, and ITRSSI.

Calling Sequence: Subroutine PRSRED (NR, NDIM, MAXNZ, JCOEF, COEF, U, VR)

Data Declaration: Integer NR, NDIM, MAXNZ, JCOEF
Real COEF, U, VR

Arguments:	NR	Number of red points.
	NDIM	Row dimension of JCOEF and COEF arrays in calling routine.
	MAXNZ	Maximum number of non-zeros per row.
	JCOEF	Integer data structure for coefficient columns.
	COEF	Data structure for array coefficients.
	U	Latest estimate of solution.
	VR	On input -- Contains the right hand side. On output -- Contains FR*UB + CR.

5.3.3.29 Subroutine PSTOP

PSTOP performs a test to see if the iterative method has converged to a solution inside the error tolerance, zeta. PSTOP is called by the subroutines ITJCG, ITJSI, ITSOR, ITSRCG, ITSRSI, ITRSCG, and ITRSSI.

Calling Sequence: Subroutine PSTOP (N, U, DNRM, CCON, IFLAG, Q1)

Data Declaration: Integer N, IFLAG
Real U, DNRM, CCON, Q1

Arguments:

N	Order of system.
U	Present solution estimate.
DNRM	Inner product of pseudo-residuals at preceding iteration.
CON	Stopping test parameter (= CCON).
IFLAG	Stopping test integer flag. = 0, SOR iteration zero. = 1, non-RS method. = 2, RS method.
Q1	Stopping test logical flag.

Common Blocks: ITCOM1
ITCOM2
ITCOM3

5.3.3.30 Subroutine SBELM

SBELM is designed to remove rows of the matrix for all off-diagonal elements are very small (less than TOL). This is to take care of matrices arising from finite element discretizations of partial differential equations with Dirichlet boundary conditions. Any such rows. corresponding columns are then eliminated (set to the identity after correcting the RHS). This routine is to be called after the subroutine SCAL by the subroutines JCG, JSI, SOR, SSORCG, SSORSI, RSCG, and RSSI.

Calling Sequence: Subroutine SBELM (N, NDIM, MAXNZ, JCOEF, COEF, RHS, WORK, TOL)

Data Declaration: Integer N, NDIM, MAXNZ, JCOEF
Real COEF, RHS, WORK, TOL

Arguments:

N	Dimension of matrix.
NDIM	Row dimension of arrays JCOEF and COEF in calling program.
MAXNZ	Maximum number of nonzero entries per row.
JCOEF	Integer array of matrix representation.
COEF	Array of sparse matrix representation.
RHS	Right hand side of matrix problem.
WORK	Work array of length N.
TOL	Tolerance factor.

5.3.3.31 Subroutine SCAL

SCAL scales original matrix to a unit diagonal matrix. RHS and U vectors are scaled accordingly. The data structure is adjusted to have diagonal entries in column 1. Zero entries in JCOEF array are changed to positive integers between 1 and N. SCAL is called by the subroutines JCG, JSI, SOR, SSORCG, SSORSI, RSCG, and RSSI.

Calling Sequence: Subroutine SCAL (N, NDIM, MAXNZ, JCOEF, COEF, RHS, U, WORK, IER)

Data Declaration: Integer N, NDIM, MAXNZ, JCOEF, IER
Real, COEF, RHS, U, WORK

Arguments:

N	Dimension of matrix.
NDIM	Row dimension of arrays JCOEF and COEF in the calling program.
MAXNZ	Maximum number of non-zero entries per row.
JCOEF	Integer array of matrix representation.
COEF	Array of sparse matrix representation.
RHS	Right hand side of matrix problem.
U	Latest estimate of solution.
WORK	Work array of length N.
IER	Error flag -- on return, nonzero values mean. 401 -- Zero diagonal element. 402 -- Nonexistent diagonal element.

Common Blocks: ITCOM1
ITCOM2
ITCOM3

5.3.3.32 Subroutine SORSCL

SORSCL scales the matrix and RHS vector by FACTOR. SORSCL is called by the subroutines SSORCG, SSORSI, ITSOR, ITSRCG, and ITSRSI.

Calling Sequence: Subroutine SORSCL (N, NDIM, MAXNZ, COEF, RHS, FACTOR)

Data Declaration: Integer N, NDIM, MAXNZ , FACTOR
Real, COEF, RHS

Arguments:

N	Dimension of matrix.
NDIM	Row dimension of array COEF.
MAXNZ	Maximum number of non-zero entries per row.
COEF	Array of sparse matrix representation.
RHS	Right hand side of matrix problem.
FACTOR	Scaling factor.

5.3.3.33 Subroutine SORWAV

SORWAV determines the wave front reordering for a forward and back SOR pass. The permutation vector corresponding to a wave front ordering is constructed. The matrix and RHS are permuted accordingly. SORWAV is called by the subroutines SOR, SSORCG, and SSORSI.

Calling Sequence: Subroutine SORWAV (N, NDIM, MAXNZ, JCOEF, COEF, RHS, U, P, IP, IWORK, WORK, NUMWAV, ICALL)

Data Declaration: Integer N, NDIM, MAXNZ, JCOEF, IP, IWORK, NUMWAV, ICALL
Real, COEF, RHS, U, P, WORK

Arguments:

N	Order of system.
NDIM	Row dimension of arrays JCOEF and COEF in the calling routine.
MAXNZ	Maximum number of non-zero entries per row.
JCOEF	Integer array for data.
COEF	Array for data structure coefficients.
RHS	Right-hand-side.
U	Solution vector.
P	Permutation vector.
IP	Inverse permutation vector.
IWORK	Integer workspace of length N.
WORK	Real workspace of length N.
NUMWAV	Number of wave fronts.
ICALL	Switch for calling. = 1 for initial call. = 2 for second call.

5.3.3.34 Subroutine UNSCAL

UNSCAL reverses the scaling done in routine SCAL. UNSCAL is called by the subroutines JCG, JSI, SOR, SSORCG, SSORSI, RSCG, and RSSI.

Calling Sequence: Subroutine UNSCAL (N, NDIM, MAXNZ, JCOEF, COEF, RHS, U, WORK)

Data Declaration: Integer N, NDIM, MAXNZ, JCOEF
Real COEF, RHS, U, WORK

Arguments:

N	Order of system.
NDIM	Row dimension of arrays JCOEF and COEF in the calling routine.

MAXNZ	Maximum number of nonzero entries per row.
JCOEF	Integer array for data.
COEF	Array for data structure coefficients.
RHS	Right hand side of matrix problem.
U	Latest estimate of solution.
WORK	Work array of length N.

5.3.3.35 Subroutine YPASX2

YPASX2 does the loop

```
      do 20 J = 1,M
        do 10 I = 1,N
          Y(I) = Y(I) + A(I,J)*X(JA(I,J))
10      continue
20      continue
```

YPASX2 is called by the subroutines PARSI and PMULT.

Calling Sequence: Subroutine YPASX2 (NDIM, N, M, A, JA, Y, X)

Data Declaration:
Integer NDIM, N, M, JA
Real A, Y, X

Arguments:

NDIM	Row dimension of A and JA arrays.
N	Order of system.
M	Number of columns in A and JA arrays.
A	Real array of active size N by M.
JA	Integer array of active size N by M.
Y	Accumulation vector.
X	Right-hand-side vector.

5.3.3.36 Subroutine YMASX2

YMASX2 does the loop

```
      do 20 j = 1,m
        do 10 i = 1,n
          y(i) = y(i) - a(i,j)*x(ja(i,j))
10    continue
20  continue
```

YMASX2 is called by the subroutines PBSOR, PFSOR, PFSOR1, PJAC, PRSBLK, and PRSRED.

Calling Sequence: Subroutine YMASX2 (NDIM, N, M, A, JA, Y, X)

Data Declaration: Integer NDIM, N, M, JA
Real A, Y, X

Arguments:

NDIM	Row dimension of A and JA arrays.
N	Order of system.
M	Number of columns in A and JA arrays.
A	Real array of active size N by M.
JA	Integer array of active size N by M.
Y	Accumulation vector.
X	Right-hand-side vector.

5.3.3.37 Subroutine VFILL

VFILL fills a vector, V, with a constant value, VAL.

Calling Sequence: Subroutine VFILL (N, V, VAL)

Data Declaration: Integer N, V, VAL

Arguments:

N	Integer length of vector V.
V	Vector.
VAL	Constant that fills first N locations of V.

5.3.3.38 Subroutine VOUT

VOUT effects printing of residual and solution vectors. It is called from PERROR.

Calling Sequence: Subroutine VOUT (N, V, ISWT, NOUT)

Data Declaration: Integer N, ISWT, NOUT
Real V

Arguments:

N	Order of system.
V	Vector of length N.
ISWT	Labelling information.
NOUT	Output device number.

5.3.3.39 Subroutine ZBRENT

ZBRENT finds the zero of a function that changes sign in a given interval (Brent algorithm). ZBRENT is called by the subroutine ECHALL.

Calling Sequence: Subroutine ZBRENT (N,TRI,EPS,NSIG,A,B,MAXFN,IER)

Data Declaration: Integer N, NSIG, MAXFN, IER
Real TRI, EPS, A, B

Arguments:

TRI	A tridiagonal matrix of order N.
EPS	First convergence criterion (input). A root, B, is accepted if ABS(F(B)) is less than or equal to EPS. EPS may be set to zero.
NSIG	Second convergence criterion (input). A root, B, is accepted if the current approximation agrees with the true solution to NSIG significant digits.
A, B	On input, the user must supply two points, A and B, such that F(A) and F(B) are opposite in sign. On output, both A and B are altered. B will contain the best approximation to the root of F.
MAXFN	On input, MAXFN should contain an upper bound on the number of function evaluations required for convergence. On output, MAXFN will contain the actual number of function evaluations used.
IER	Error parameter. (output). Terminal error. IER = 501 indicates the algorithm failed to converge in MAXFN evaluations. IER= 502 indicates F(A) and F(B) have the same sign.

Common Blocks: ITCOM1
ITCOM2
ITCOM3

5.3.3.40 Subroutine WHENIGE

WHENIGE is called by the subroutine PRBNDX.

Calling Sequence: Subroutine WHENIGE (N, P, INC, ITARG, IP, NPT)

Data Declaration: Integer N, INC, ITARG, IP, NPT
Real P

Arguments:

N	Order of system.
P	
INC	
ITARG	
IP	
NPT	

5.3.3.41 Subroutine WHENILT

WHENILT is called by the subroutine PRBNDX.

Calling Sequence: Subroutine WHENILT (N, P, INC, ITARG, IP, NPT)

Data Declaration: Integer N, INC, ITARG, IP, NPT
Real P

Arguments:

N	Order of system.
P	Permutation vector.
INC	
ITARG	
IP	
NPT	

5.3.3.42 Subroutine SAXPY

SAXPY overwrites single precision SY with single precision SA*SX +SY. SAXPY is called by the subroutine PFSOR1.

Calling Sequence: Subroutine SAXPY(N, SA, SX, INCX, SY, INCY)

Data Declaration: Integer N, INCX, INCY
Real SA, SX, SY

Arguments:

N	Order of system.
SA	
SX	
INCX	
SY	
INCY	

5.3.3.43 Subroutine SCOPY

SCOPY copies single precision SX to single precision SY. SCOPY is called by the subroutines JCG, JSI, SSORCG, SSORSI, RSCG, RSSI, ITJSI, ITSOR, ITSRCG, ITSRSI, ITRSSI, PERMAT, PERVEC, PFSOR1, SCAL, and UNSCAL.

Calling Sequence: Subroutine SCOPY (N, SX, INCX, SY, INCY)

Data Declaration: Integer N, INCX, INCY
Real SX, SY

Arguments:

N	Order of system.
SX	
INCX	
SY	
INCY	

5.3.3.44 Subroutine SSCAL

SSCAL replaces single precision SX by single precision SA*SX. SSCAL is called by the subroutine SORSCL.

Calling Sequence: Subroutine SSCAL (N, SA, SX, INCX)

Data Declaration:
Integer N, INCX
Real SA, SX

Arguments:

N	Order of system.
SA	
SX	
INCX	

6.0 ACRONYMS AND ABBREVIATIONS

ADCIRC	Advanced Circulation for Shelves, Coastal Seas, and Estuaries
CSCI	Computer Software Configuration Item
CSC	Computer Software Component
ETA-29	Operational numerical forecast model
FD	Finite Difference
FE	Finite Element
FNMOC	Fleet Numerical Meteorology and Oceanography Center
GWCE	Generalized Wave-Continuity Equation
HPC	High Performance Computing
ITPACKV 2D	A vectorized version of ITPACK 2C for the Cray Y-MP and similar vector computers that is used for solving large sparse linear systems by iterative methods
LSQ	Least squares
ME	Momentum Equation
MET	Meteorological
NRL	Naval Research Laboratory
NWS AVN	National Weather Service, Aviation Model
PBL/JAG	Planetary Boundary Layer Atmosphere Model
PSI	Planning Systems, Incorporated
SDD	Software Design Description
SSC	Stennis Space Center
UNIX	Workstation Operating System
WSC	Warfighting Support Center

7.0 APPENDIX I. FORTRAN COMMON BLOCKS

7.1 CSC ADCIRC COMMON BLOCKS

7.1.1 COMMON / LSQFREQS/

Variable	Type	Description
FACE	Real	Equilibrium argument in degrees for tidal forcing on elevation specified boundaries.
FF	Real	Nodal factor.
FREQ	Real	Frequency (rad/s)
HAFACE	Real	Equilibrium argument (degrees).
HAFF	Real	Nodal factor.
HAFNAM	Real	An alphanumeric descriptor whose length must be <= 10 characters.
HAFREQ	Real	Frequency (rad/s).
NAMEFR	Integer	An alphanumeric descriptor whose length must be <=10 characters.
NFREQ	Integer	Number of frequencies to include in harmonic analysis of model results.
NHARFR	Integer	Number of frequencies to include in harmonic analysis of model results.

7.1.2 COMMON / HGRID/

Variable	Type	Description
DP	Real	Nodal depth array
NM	Integer	Element connectivity array
SFAC	Real	Nodal CPP conversion factor array
X	Real	Nodal X-coordinate array
Y	Real	Nodal Y-coordinate array

7.1.3 COMMON / ELEAREA/

Variable	Type	Description
AREAS(MNE)	Real	Element areas array

7.1.4 COMMON / MEANSQ/

Variable	Type	Description
DT	Real	Time step (in seconds)
FMV	Real	Fraction of the harmonic analysis period (extending back from the end of the harmonic analysis period) to use for comparing the water elevation and velocity means and variances from the raw model time series resynthesized from the harmonic constituents.

ITMV	Integer	Time step at which means and variances of the harmonic analyses are completed.
NTSTEPS	Integer	Number of time steps since harmonic analysis means and variance checking has begun.
TIMEBEG	Real	Time at which mean or variance calculation begins

7.1.5 COMMON / MEANSQE/

Variable	Type	Description
ELAV	Real	Elevation mean, i.e. sum of elevations computed by ADCIRC over all time steps since harmonic analysis means and variance checking has begun.
ELVA	Real	Elevation variance, i.e. sum of squares of elevations computed by ADCIRC over all time steps since harmonic analysis means and variance checking has begun.

7.1.6 COMMON / MEANSQV/

Variable	Type	Description
XVELAV	Real	X-component of velocity mean, i.e. sum of depth-averaged X-component velocities computed by ADCIRC.
XVELVA	Real	X-component of velocity variance, i.e. sum of squares of elevations computed by ADCIRC over all time steps since harmonic analysis means and variance checking has begun.
YVELAV	Real	Y-component of velocity mean, i.e. sum of depth-averaged Y-component velocities computed by ADCIRC over all time steps since harmonic analysis means and variance checking has begun.
YVELVA	Real	Y-component of velocity variance, i.e. sum of squares of depth-averaged V velocities computed by ADCIRC over all time steps since harmonic analysis means and variance checking has begun.

7.1.7 COMMON / RAWMET/

Variable	Type	Description
PE	Real	Surface pressure (millibars to m of water)
PG	Real	Surface pressure (N/m^2 to m of water)
PRN	Real	Applied atmospheric pressure at the free surface.
UE	Real	X-component of wind speed at 10 m (cm/s in E grid orientation).

Variable	Type	Description
UG	Real	X-component of wind speed (m/s)
VE	Real	Y-component of wind speed at 10 m (cm/s in E grid orientation).
VG	Real	Y-component of wind speed (m/s)
WVXFN	Real	Horizontal wind speed.
WVYFN	Real	Horizontal wind direction (deg.).

7.1.8 COMMON / LSQPARMS/

Variable	Type	Description
ICALL	Integer	Number of times the harmonic analysis has been updated.
ITUD	Integer	Model time step when the harmonic analysis was last updated.
MM	Integer	= 2*NHARFR – NF
NF	Integer	Indicator of whether the steady frequency is included in the harmonic analysis (NF=1, steady is included; NF=0, steady is not included).
NZ	Integer	Indicator of whether the steady frequency is included in the harmonic analysis (NZ=0, steady is included; NZ=1, steady is not included).
TIMEUD	Real	Model time when the harmonic analysis was last updated.

7.1.9 COMMON / MATRIX/

Variable	Type	Description
A	Real	Coefficient in the least squares matrix used for the harmonic analysis.
B	Real	RHS load vector.
P	Real	RHS load vector.
X	Real	Solution vector.

7.1.10 COMMON / LOADVECES/

Variable	Type	Description
STAELV	Real	Harmonic analysis load vectors for elevation at elevation recording stations.

7.1.11 COMMON / LOADVECVS/

Variable	Type	Description
STAULV	Real	Harmonic analysis load vectors for depth-averaged u velocity at velocity recording stations.
STAVLV	Real	Harmonic analysis load vectors for depth-averaged v velocity at velocity recording stations.

7.1.12 COMMON / LOADVECEG/

Variable	Type	Description
GLOELV	Real	Harmonic analysis load vectors for elevation.

7.1.13 COMMON / LOADVECVG/

Variable	Type	Description
GLOULV	Real	Harmonic analysis load vectors for depth-averaged u velocity.
GLOVLV	Real	Harmonic analysis load vectors for depth-averaged v velocity.

7.2 CSC SOLVER COMMON BLOCKS

7.2.1 COMMON / ITCOM1/

Variable	Type	Description
IN	Integer	Iteration number.
IS	Integer	Iteration number when parameters last changed.
ISYM	Integer	Symmetric/nonsymmetric case switch.
ITMAX	Integer	Maximum number of iterations allowed.
LEVEL	Integer	Level of output control switch.
NOUT	Integer	Output UNIT number.

7.2.2 COMMON / ITCOM2/

Variable	Type	Description
ADAPT	Real	Fully adaptive procedure switch.
BETADT	Real	Switch for adaptive determination of beta.
CASEII	Real	Adaptive procedure case switch.
HALT	Real	Stopping test switch.
PARTAD	Real	Partially adaptive procedure switch.

7.2.3 COMMON / ITCOM3/

Variable	Type	Description
BDELNM	Real	Two norm of B times delta-super-N.
BETAB	Real	Estimate for the spectral radius of LU matrix.
CME	Real	Estimate of largest eigenvalue.
DELNNM	Real	Inner product of pseudo-residual at iteration N.
DELSNM	Real	Inner product of pseudo-residual at iteration S.
FF	Real	Adaptive procedure damping factor.
GAMMA	Real	Acceleration parameter.
OMEGA	Real	Over-relaxation parameter for SOR and SSOR.
QA	Real	Pseudo-residual ratio.
QT	Real	Virtual spectral radius.
RHO	Real	Acceleration parameter.
RRR	Real	Adaptive parameter.
SIGE	Real	Parameter sigma-sub-E.
SME	Real	Estimate of smallest Eigen value.
SPECR	Real	Spectral radius estimate for SSOR.
SRELPR	Real	Spectral radius estimate for SSOR.
STPTST	Real	Stopping parameter.
UDNM	Real	Two norm of U.
ZETA	Real	Stopping criterion.

8.0 APPENDIX II. Supplementary Notes

8.1 NWS

NWS = 10

The wind velocity and surface pressure data files are named using the following convention:

UNIT 200 - Wind and pressure at the time of a model hot start. This file is not used for a cold start.
UNIT 206 - Wind and pressure 6 hours after a cold or hot start.
UNIT 212 – Wind and pressure 12 hours after a cold or hot start.
UNIT 218 – Wind and pressure 18 hours after a cold or hot start.
....
and so on until the end of the run.

Note: If the model is hot started, it must be done at an even 6 hour interval so that the hot start time corresponds to the time of a wind/pressure field file. The wind/pressure files must also be renamed so that they begin with UNIT 200 at the time of the hot start and continue from there. Enough wind/pressure files must be present to extend through the ending time of the model run.

The first data set must be 6 hours after the beginning of a cold start (UNIT 206) or at the beginning of a hot start (UNIT 200)

Data must be provided for the entire model run, OTHERWISE THE RUN WILL CRASH!!!

The following transformations are preformed to put this info into usable form for the model calculations:

```
WIND_STRESS = DRAG_COEFF*0.001293*WIND_VEL*WIND_SPEED
DRAG_COEFF = 0.001*(0.75+0.067*WIND_SPEED)
IF(DRAG_COEFF.GT.0.003) DRAG_COEFF=0.003
PR2/GRAVITY/1000.
```

NWS = 11

The NWS ETA-29 MET files are in binary and have the format described below. Each file is assumed to contain one day's worth of data (8 data sets, one every 3 hours, beginning at 0300 and continuing through 2400 of the given day) on the ETA grid. The files are named using the following convention:

UNIT 200 - Wind and pressure the day before a model run is hot started. The final data in this file are used as the initial MET condition for the hot start. This file is not used for a cold start.
UNIT 201 – Wind and pressure during Day 1 of the run or Day 1of a hot start.
UNIT 202 – Wind and pressure during Day 2 of the run or Day 2 of a hot start.
UNIT 203 - Wind and pressure during Day 3 of the run or Day 3 of a hot start.
....
and so on until the end of the run.

Note: The wind data is converted to an east-west, north-south coordinate system inside ADCIRC. If the model is hot started, it must be done at an even day interval so that the hot start time corresponds to the time of a wind/pressure field file. Also, the wind/pressure files must be renamed so that they begin with UNIT 200 at the time of the hot start and continue from there. Enough wind/pressure files must be present to through the ending time of the model run.

The first data set must be 3 hours after the beginning of a Cold start (UNIT 201) or at the beginning of a hot start (UNIT 200).

Data must be provided for the entire model run, OTHERWISE THE RUN WILL CRASH!!!

The following transformations are preformed to put this information into usable form for the model calculations:
WIND_STRESS = DRAG_COEFF*0.001293*WIND_VEL*WIND_SPEED
DRAG_COEFF = 0.001*(0.75+0.067*WIND_SPEED)
IF(DRAG_COEFF.GT.0.003) DRAG_COEFF=0.003
PR2/GRAVITY/1000.

8.2 FMV

Notes: The means and variance calculations are only done if the global harmonic calculations are performed. Results are written out to UNIT 55. A summary of the poorest comparisons throughout the domain and the node numbers where these occurred is given at the end of the UNIT 6 output file.

The time series resynthesis from the harmonic constituents can use up a lot of CPU time since this is done for every time step during the specified part of the harmonic analysis period. If the harmonic analysis period extends for only a few days, it is practical to set up FMV = 1. Otherwise, compute means and variances for more than 10 – 20 days. Ultimately, the practical limit to these calculations depends on the number of nodes, the number of constituents in the harmonic analysis, and the size of the time step.

8.3 IBTYPE

IBTYPE = 3, 13, 23

Notes: Outward normal flow per unit width at an external barrier boundary node I is computed as:

```
IF(ETA2(NNBB).GT.BARLANHT(I))
  QN2(I) = -(2./3.)*BARLANCFSP(I)
           *(ETA2(NNBB)-BARLANHT(I))*
           *((2./3.)*(ETA2(NNBB)-BARLANHT(I))*G)**0.5
IF(ETA2(NNBB).LE.BARLANHT(I))
  QN2(I) = 0
```

Where

NNBB = NBVV(K,I)	Global node number for external barrier boundary node
ETA2(NNBB)	The known time level surface elevation solution at the external barrier boundary node as computed in the code.

This formula is given by Leendertse (Aspects of SIMSYS2D - A system for two-dimensional flow computation, RAND/R-3572-USGS, 1987) and is simply the formula for a broad crested weir (e.g. see Henderson, open channel flow, Section 6.6). Also see Appendix III.

IBTYPE = 4, 24 (Internal Barrier Type Boundaries)

Notes: Cross barrier normal flow per unit width at internal barrier boundary node I is computed as:

Case 1 - Water level below barrier. Occurs if ((RBARWL1.LT.0.0).AND.(RBARWL2.LT.0.0)). Results in no flow across the barrier (QN2(I)=0.0).

Case 2 - Water level equal on both sides of barrier. Occurs if (RBARWL1.EQ.RBARWL2). Results in no flow across the barrier (QN2(I)=0.0).

Case 3 - Water level greater on front side of the barrier but elevation difference such that cross barrier flow is subcritical. Occurs if ((RBARWL1.GT.RBARWL2).AND.(RBARWL1.GT.0.0).AND. (RBARWL2.GT.RBARWL1F)). Results in subcritical normal flow across the barrier from front to back (QN2(I)=-RAMP*BARINCFSB(I)*RBARWL2* (2*G*(RBARWL1-RBARWL2))**0.5).

Case 4 - Water level greater on front side of the barrier but elevation difference such that cross barrier flow is supercritical. Occurs if ((RBARWL1.GT.RBARWL2).AND.(RBARWL1.GT.0.0).AND.

(RBARWL2.LE.RBARWL1F)). Results in supercritical normal flow across barrier from front to back (QN2(I)=-RAMP*BARINCFSP(I)*RBARWL1F*(RBARWL1F*G)**0.5).

Case 5 - Water level lower on front side of the barrier but elevation difference such that cross barrier flow is subcritical. Occurs if ((RBARWL2.GT.RBARWL1).AND.(RBARWL2.GT.0.0).AND.(RBARWL1.GT.RBARWL2F)). Results in sub-critical normal flow across the barrier from back to front (QN2(I)=RAMP*BARINCFSB(I)*RBARWL1 (2*G*(RBARWL2-RBARWL1))**0.5).

Case 6 - Water level lower on front side of the barrier but elevation difference such that cross barrier flow is supercritical. Occurs if ((RBARWL2.GT.RBARWL1).AND.(RBARWL2.GT.0.0).AND.(RBARWL1.LE.RBARWL2F)). Results in supercritical normal flow across the barrier from back to front (QN2(I)=RAMP*BARINCFSP(I)*RBARWL2F*(RBARWL2F*G)**0.5).

Where

ETA2(J) The known time level surface elevation solution at global node J computed in the code.

NNBB1 = NBVV(K,I) Global node number on the front side of the barrier.

NNBB2 = IBCONN(I) Global node number on the back side of the barrier.

RBARWL1 = ETA2(NNBB1)-BARINHT(I)
Relative water height on the front side of the barrier.

RBARWL2 = ETA2(NNBB2)-BARINHT(I)
Relative water height on the back side of the barrier.

RBARWL1F=2.0*RBARWL1/3.0

RBARWL2F=2.0*RBARWL2/3.0

Note that the flow on the back side of the barrier is set equal and opposite to the flow on the front side of the barrier. These formulae are given by Leendertse (Aspects of SIMSYS2D - A system for two-dimensional flow computation, RAND/R-3572-USGS, 1987) and are simply the formulae for a broad crested weir (e.g. see Henderson, open channel flow). Also see Appendix III.

8.4 ITITER, ISLDIA, CONVCR, ITMAX

Parameters that provide information about the solver that will be used for the GWCE. (Note: All of the parameters must be input regardless of whether a direct or iterative solver is selected. However, ISLDIA, CONVCR and ITMAX are only used with the iterative solvers) Note: The best luck has been in using the JCG solver (the JSI also works ok). The others either blow up rapidly or don't work because the grid can not be red/black ordered. Typically use CONVCR=1e-6 on the CRAY and CONVCR=2.98e-5 on a 32 bit machine. After the first few time steps, the solutions usually converge within 5-10 iterations.

9.0 APPENDIX III. General Notes for Normal Flow Boundary Condition Specification

- All external (external no normal flow, external with specified normal flow and external barrier) boundary segments should be listed in consecutive order around the outside of the entire domain before any internal (island with no normal flow or internal barrier) boundary segments are listed.
- All no normal flow internal boundaries should be closed by repeating the first node as the last node.
- An external boundary that completely surrounds the domain (e.g., a lake) should be closed by repeating the first node as the last node.
- Unless the boundary segment is closed, always start a boundary segment where two different types of boundaries meet.
- At a node where an external specified normal flow or an external barrier boundary meets an external no normal flow boundary, the initial leg of the external specified normal flow boundary or external barrier boundary is used to determine the normal and tangential direction.
- External boundaries with specified (non-zero) normal flow boundary conditions and external barrier boundaries must be separated by an external no normal flow boundary segment or an elevation specified boundary segment (i.e. External boundaries with specified (non-zero) normal flow boundary conditions and external barrier boundaries segments can not overlap).
- Internal barrier boundaries can overlap external no normal flow boundary segments. In this case the external no normal flow nodes may be treated in the weak sense and set equal to zero in the GWCE flow integral only, while the internal barrier overlapping nodes are handled as indicated in the input file. Specifically, the following cases are checked for:
    1) The external no flow boundary node is specified as essential with slip (IBTYPE=0) and the internal barrier boundary node is specified as essential with slip (IBTYPE=4). The code resets the external no flow boundary node to be a natural no flow boundary node (IBTYPE=20).
    2) The external no flow boundary node is specified as essential with no slip (IBTYPE=10) and the internal barrier boundary node is specified as essential with slip (IBTYPE=4). The code resets the external no flow boundary node to be a natural no flow boundary node (IBTYPE=20).
    3) The external no flow boundary node is specified as natural with slip (IBTYPE=20) and the internal barrier boundary node is specified as essential with slip (IBTYPE=4). The code takes no action to change user input.
    4) The external no flow boundary node is specified as essential with slip (IBTYPE=0) and the internal barrier boundary node is specified as natural with slip (IBTYPE=24). The code takes no action to change user input.
    5) The external no flow boundary node is specified as essential with no slip (IBTYPE=10) and the internal barrier boundary node is specified as natural with slip (IBTYPE=24). The code resets the external no flow boundary node to be an essential no flow with slip boundary node (IBTYPE=0).

6) The external no flow boundary node is specified as natural with slip (IBTYPE=20) and the internal barrier boundary node is specified as natural with slip (IBTYPE=24). The code takes no action to change user input.

- Internal barrier boundaries can not overlap external specified flow boundary segments, external barrier boundary segments or internal no normal flow boundaries.
- For all normal flow boundaries (i.e. IBTYPE = 0, 1, 2, 3, 4, 10, 11, 12, 13, 20, 21, 22, 23, 24, 30), the flow boundary integral in the GWCE is evaluated with the appropriate (zero, specified or computed) flow. This is a natural boudary condition implementation for some normal flow boundaries (i.e. IBTYPE = 0, 1, 2, 3, 4, 10, 11, 12, 13). The normal direction velocity component is set equal to the appropriate (zero, specified, or computed) flow value by eliminating the normal direction momentum equation (obtained after re-orienting the x-y momentum equations into n-t directions). This momentum equation is replaced with the normal flow value on the boundary expressed in terms of velocity (dividing the normal flow by the actual total water column height). This is an essential boundary condition implementation that also includes the previously described natural boundary condition implementation
- For some normal flow boundaries (i.e. IBTYPE = 10, 11, 12, 13), the tangential direction velocity component is set equal to zero by eliminating the tangential direction momentum equation and replacing it with a zero prescribed tangential velocity component. This is an essential boundary condition implementation that also includes the previously described natural and essential normal flow boundary condition implementations. Use of this boundary condition implementation requires considerable care since this type of no slip boundary condition is only mathematically justifiable if lateral viscous terms are used in the simulation and only physically justifiable if the lateral boundary layers are sufficiently resolved.